Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

Non-Alcoholic Fatty Liver Early Prediction Using Squared Reinforcement Learning Based on Primary Biliary Cirrhosis

Milad Tarighat[1,2], Javad Mohammadzadeh[1,2*], Hadi Saboohi[1,2], Alireza Nikravanshalmani[1,2]

[1] Department of Computer Engineering, Ka. C., Islamic Azad University, Karaj, Iran
[2] Institute of Artificial Intelligence and Social and Advanced Technology, Ka. C., Islamic Azad University, Karaj, Iran
E-mail: Javad.mohammadzadeh@iau.ac.ir

Received: 19 December 2024; **Revised:** 17 March 2025; **Accepted:** 24 March 2025

Abstract: Biomedical engineering and artificial intelligence encompass many intricate challenges worthy of investigation. An essential challenge within this domain is identifying an effective classifier algorithm for predictive analytics. This challenge holds significant importance, primarily due to the extensive time it often requires for resolution. Hence, developing an algorithm that autonomously identifies optimal classification methods becomes vital. Classification algorithms play a pivotal role in diagnosing and forecasting various health-related issues. Leveraging artificial intelligence to predict human diseases is particularly beneficial among various applications. A prominent example of this application involves the prediction of primary biliary cirrhosis utilizing classification algorithms. This research presents a reinforcement learning framework designed to autonomously acquire the most effective classification and balancing algorithms for predicting this disease. The proposed framework draws inspiration from a voting mechanism. This approach is designated as Square Reinforcement Learning (SRL) by integrating four distinct classification metrics. The findings of this study demonstrate that the proposed SRL method enhanced the performance of classification algorithms, increasing accuracy from 85% to an impressive 98%.

Keywords: reinforcement learning, classification, square learning, primary biliary cirrhosis, feature selection

MSC: 65L05, 34K06, 34K28

1. Introduction

Nowadays, fatty liver disease and its negative repercussions are a problem for many people. Liver disorders kill almost two million people annually [1]. However, applying machine learning algorithms to avoid fatty liver at an early stage is widespread. One of the leading causes of this illness is fatty liver, which can be either alcoholic or non-alcoholic [2]. Primary biliary liver disease is one of the non-alcoholic chronic liver disorders [3]. As a chronic condition, it gradually impacts a person's life until liver cancer develops in the individual [4]. Therefore, it is crucial to prevent this disease.

A doctor oversees a medical treatment plan for Primary Biliary Cholangitis (PBC) at home or a hospital. This condition does not have a specific therapy; thus, doctors treat it as a chronic condition by administering medication [5]. Utilizing Ursodiol, which aids in the liver's bile removal, is one of the workable options [6]. The liver function test is

DOI: https://doi.org/10.37256/cm.6320256275

a must for all medicinal solutions. Machine learning techniques are, therefore, desperately needed in this discipline to diagnose patients. Although machine learning systems have some predictive ability, more is needed [7].

The performance of each machine learning algorithm varies from that of the others. These problems are addressed by researchers employing supervised algorithms (classification) for prediction [8]. It is impossible to employ each supervised learning method individually because there are so many of them [9]. Researchers create and present new algorithms daily that are more likely to perform better than older versions and the opposite [9]. These algorithms have applications in several other fields, including medical [10], improved learning algorithm selection [11], spotting extra data in the network [12], and multimedia [13]. There is still a fundamental issue with choosing algorithms, even assuming that the necessary data is appropriate for the algorithms and that the algorithms operate flawlessly.

In this research, an algorithm that can learn has been presented. This learning will help select the best classification system for diagnosing PBC disease. A new framework is implemented to determine which algorithm is better. Reinforcement learning techniques are the idea for this system. After finding inspiration, this framework's policies were put into practice, and the best algorithm was chosen from various algorithms. The algorithm has demonstrated noteworthy performance, and this research is unusual in that it builds a reinforcement learning framework. These are a few of this article's significant contributions to medicine and machine learning.

- Determining the crucial requirement for machine learning-based fatty liver prediction.
- Addressing the challenge of selecting optimal classifiers for imbalanced datasets.
- Demonstrating the applicability of reinforcement learning to medical diagnosis.
- Providing a novel method for early prediction of PBC, which can be adapted to other diseases.

The second part of this study project studies primary hepatobiliary illness and emphasizes its significance. The final section then provides the new solution and explains how each component works. The fourth section follows with a discussion of the new method's test and evaluation outcomes. The fifth and final section concludes and discusses potential future research in this area.

2. Literature review

The liver is harmed by primary biliary cirrhosis. This illness is chronic, which means that it lasts a long time or frequently recurs [4]. People with primary biliary cholangitis have damaged bile ducts [14]. The liver has microscopic tubes called bile ducts that carry bile, a chemical required for digestion, to other bodily regions [15]. Cirrhosis is the medical term for liver ulcers that result from bile buildup [16]. Since primary biliary liver disease is a progressive disorder, the patient's condition worsens over time. Untreated liver cirrhosis, liver failure, and even death can result [17].

Early-stage PBC patients frequently have no symptoms [18]. Some patients learn they have it when their doctor examines them for another issue [18]. Tiredness, skin itchiness (pruritus), abdominal pain, dark skin, tiny white or yellow lumps under the skin around the eyes, dry eyes and mouth, and pain in the muscles and joints are just a few instances of the general symptoms of this illness [16]. Any other illness could cause the symptoms mentioned earlier, but as this liver condition worsens, additional particular symptoms emerge. Jaundice, which is characterized by yellow skin and white eyes, swelling of the feet, knees, and legs, ascites, internal bleeding in the upper portion of the stomach or lower esophagus produced by large arteries, nausea, weight loss, and black urine are signs of PBC in its later stages [19].

Doctors typically inquire about patients' and their families' histories to identify this illness [20]. For precise diagnosis, examinations like blood tests and ultrasounds are occasionally employed [21]. These exams take time and cost money and time. For these reasons, the field of machine learning, known as artificial intelligence, has its roots in medical assistance. There are many datasets available today for PBC disease [22]. Additionally, numerous algorithms can be used to learn about this condition from data [23]. This research aimed to provide a framework for an automatic and precise disease diagnosis.

Information and data are the first prerequisites for proposing such predictive systems [24]. Because of this, this article uses a trustworthy dataset related to PBC disease. For this reason, the Mayo Clinic study in Primary Biliary Cirrhosis (PBC) of the liver [25], an innovative dataset in the field of PBC disease, is included in this article. Four

hundred twenty-four patients who visited the Mayo Clinic over ten years, from 1974 to 1984, are included in this dataset [26]. The recording of fatalities after the inclusion of patients, which can be used for time series analysis, is one of the dataset's crucial components. Twenty medical and clinical features make up the Mayo dataset, which makes it an excellent candidate for machine learning techniques. The research has been examined in greater detail in the sections that follow.

The proposed Square Reinforcement Learning (SRL) framework introduces several novel contributions that distinguish it from existing methods in disease prediction. Unlike traditional approaches that rely on predefined algorithms or static optimization techniques, the SRL framework autonomously selects the most effective classification and balancing algorithms using a reinforcement learning mechanism. This autonomy, combined with its dynamic adaptability to imbalanced data and varying dataset characteristics, ensures robust performance across all classes. The framework achieves an accuracy of 98% in predicting Primary Biliary Cholangitis (PBC), significantly outperforming traditional methods. Furthermore, its modular design and computational efficiency make it suitable for real-time clinical applications and adaptable to other medical conditions. These contributions position the SRL framework as a versatile and effective tool for medical diagnosis and prediction.

3. Square reinforcement learning (SRL)

A reinforcement learning method is provided in this study for the early diagnosis of PBC disease. This process, based on a voting mechanism, has three essential components in Figure 1. The dataset is organized and prepared for machine learning techniques to be applied in the first section of this framework. The reinforcement learning approach motivated by voting systems begins to function in the second portion. Finally, learning is done to select the classification algorithm in the third section of the framework, and prediction is also shown. Figure 2 depicts the proposed method's overview.

This section is divided into three subsections. The initial stage of Square Reinforcement Learning (SRL) is described in part 3.1. The SL method, which is the primary subject of this article, is thoroughly detailed in Section 3.2. The execution of selecting the optimal feature selection technique is then discussed in Section 3.3. The framework's final step is described in Section 3.4, along with the algorithm and flowchart.

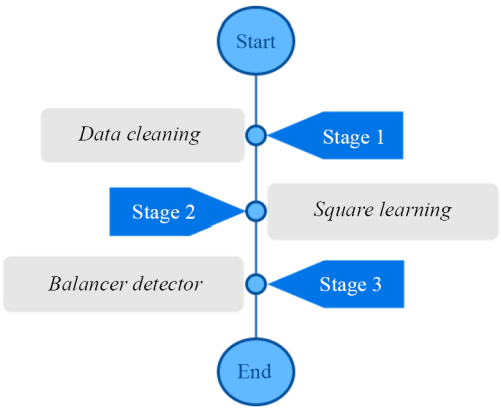

Figure 1. Overview of the SRL method

3.1 *Data cleaning*

The dataset is subjected to four critical tasks in the first stage of this framework. In addition to privacy detection, these procedures comprise null, duplicate, and outlier detection. Here is a detailed description of each.

3.2 *Privacy detection*

Since it was intended for this framework to function with a variety of datasets, any personal information like a national identification number, address, phone number, or similar items will be removed from the datasets in this part. Additionally, superfluous features like patient IDs will be deleted. Additionally, data figures are retrieved to analyze the features visually.

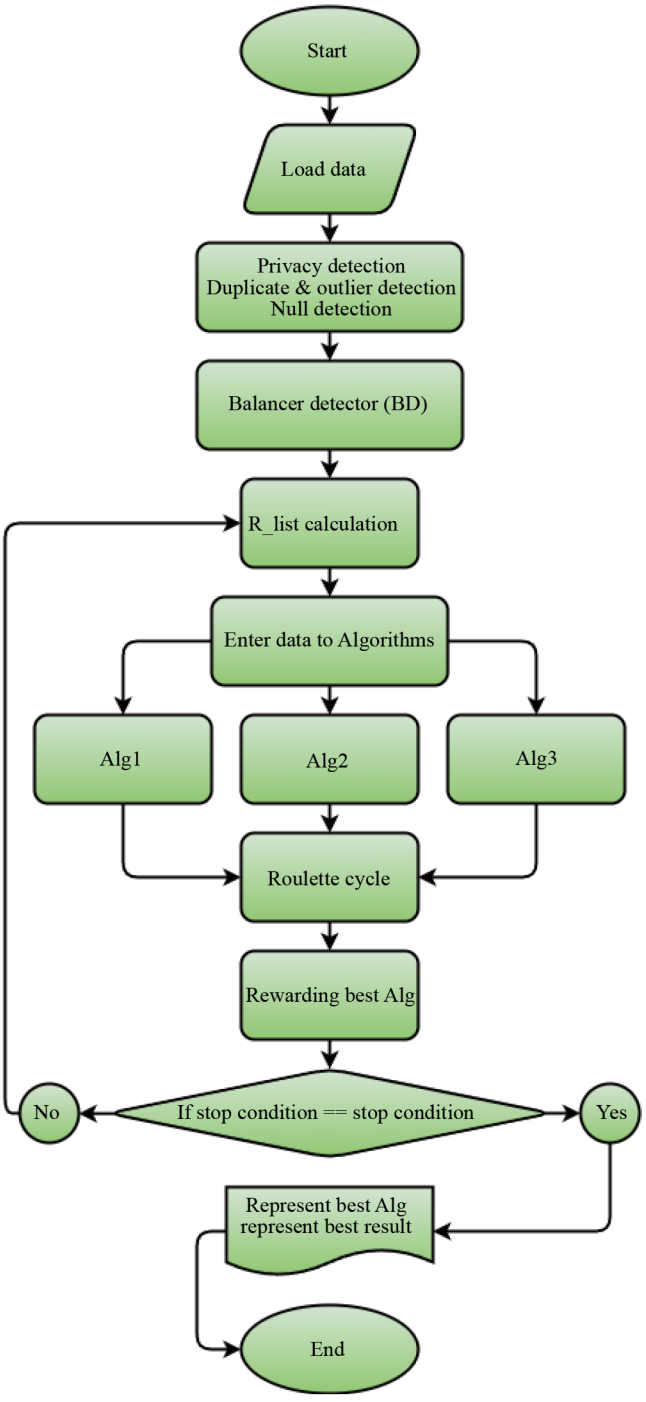

Figure 2. Flowchart of SRL

3.3 *Duplicate & outlier detection*

At this stage, duplicate samples have been extracted first to get data ready for usage. The features that have outliers have also been identified after being extracted. After that, these two issues are integrated, eliminating duplicate and excellent examples.

3.4 *Null detection*

Real datasets have a high likelihood of containing inaccurate data. For this reason, all inaccurate data have been removed from this area. Once the data balance is verified, the available null data will be removed. If the data set and its objective are out of balance, balancing techniques regulate the data before it moves on to the next phase. Some well-known methods have been applied in this sector to improve the performance of classifiers and balancers. For this balance, the Shark Smell Optimization (SSO) technique is employed [27].

3.5 *Square learning (SL)*

Number this part contains the core of the research's structure and its originality. The best classifier selection issue is so complicated that the system can comprehend it independently. The issue of the best algorithm can be resolved using reinforcement learning techniques [10]. This study suggests a reinforcement learning approach to address this issue. The framework's work then starts with integrating the dataset, classification, and voting algorithms.

All classification algorithms are run once the data from the initial section of the framework has been removed. The fitting algorithm is then chosen using the N-input Voting technique. The assessment and scoring system are then put into practice. The algorithms must first be executed. The Decision Tree algorithm (DT) [28], Random Forest algorithm (RF) [29], and Support Vector Classifier (SVC) algorithm [30] are the algorithms used in this section.

There is a list of algorithms in this algorithm called AlgL. This list's elements are shown as $AlgL = Alg_1$, Alg_2, $Alg_3, \ldots, Alg_n$. All classifiers must be run once in this algorithm. This initial implementation is carried out since they are subjected to initial scoring. This reinforcement learning algorithm's policy is to acquire the best possible score. Equation (1) is used to establish the score of each algorithm. The score is equal to the Square Learning rate in this aspect. Precision, recall, F-major, and accuracy measurements were employed to comprehend SL better, where Pr denotes precision, R denotes recall, Fs denotes F major, and Acc denotes accuracy.

$$SL = \frac{\sum_{i=1}^{\infty} (PR \cdot RC \cdot ACC \cdot F1)}{100} \tag{1}$$

SL is determined for each algorithm and kept in a list called the Algorithm Reward ($AlgR$) in the order of the most significant mean of this parameter to the minimum. The essential element is that all algorithms must have a score in each Iteration. As a result, Ci and I are introduced as parameters for current iteration and overall iteration. In addition, Equation (2) specifies the scoring requirement for each method. In Equation (2), the score belongs to Ward's method if it has the maximum value of SL and is in the current iteration (Ci). The scoring should be updated as the algorithm's iteration phases continue. Algorithm Rewards ($AlgR$) are kept in a list called the Reward List ($RList$), which is updated after each iteration. The max square learning rate (Alg_{Max}^{SL}) parameter indicates the optimal algorithm with the most fantastic score. The total scores of each method are saved in the list of variables added in Equations (3) and (4).

$$AlgR = \{AlgR + 1 \mid Alg_{SL} = \text{Maximum},\ \text{Iteration} = Ci\} \tag{2}$$

$$AlgR = \sum_{i=1}^{n} (Alg_{Max}^{SL} + 1) \tag{3}$$

$$RList = \{\text{List } Alg | \text{Maximum } SL > Alg > \text{Minimum } SL\} \tag{4}$$

In order to select a new algorithm, the *RList* ranks the algorithms from best to worst. To prevent the greedy selection of the best algorithm utilizing a roulette wheel in this algorithm, all algorithms can be picked until the last step of repetition. The *RW* parameter, which stands for roulette wheel, introduces roulette wheel output. Equation (5) depicts a roulette wheel. The best algorithm is divided by the total scores of the other algorithms in this calculation. This technique also offers weak algorithms a chance. The final stage is final learning, which will be chosen and executed following the algorithm output process from the roulette wheel, the order of which is indicated in Equation (6). *Ai* indicates the action for each iteration and is equivalent to the roulette cycle execution of the specified algorithm. This procedure continues till the criterion is met. The number of iterations is the criteria for halting in this framework.

$$RW = \frac{Alg_{Max}^{SL}}{\sum_{Alg=1}^{n} Alg_{SL}} \tag{5}$$

$$Ai = \left\{ \text{Run selected } Alg \middle| \frac{Alg_{Max}^{SL}}{\sum_{Alg=1}^{n} Alg_{SL}} \right\} \tag{6}$$

The SRL framework is based on a voting mechanism that combines four performance metrics: precision Pr, recall Re, F1-score $F1$, and accuracy Acc. The reward function for each classifier i is given by Equation (7).

$$R_i = \sum_{t=1}^{\infty} (Pr_t \times Re_t \times F1_t \times Acc_t) \tag{7}$$

where t denotes the time step in the reinforcement learning process.

The algorithm ranks classifiers using a roulette wheel selection method to avoid greedy selection. The probability of selecting a classifier is proportional to its reward.

The convergence of the SRL framework can be established by analyzing the reward function and the iterative nature of the algorithm. The reward function, which combines precision, recall, F1-score, and accuracy, ensures that the algorithm progressively selects classifiers that maximize these metrics. By leveraging the properties of reinforcement learning, we can demonstrate that the SRL framework converges to an optimal classifier selection strategy. Specifically, the iterative process of updating the Reward List (*RList*) and the roulette wheel selection mechanism ensures that the algorithm explores and exploits the classifier space effectively, leading to convergence. A formal proof of convergence is provided in Equation (7), where we outline the mathematical foundations of the SRL framework and its convergence properties.

We demonstrate the convergence of the Square Reinforcement Learning (SRL) framework in more detail. For this purpose, we use basic reinforcement learning concepts such as the Bellman equation and Markov Decision Processes (MDP). The value function is calculated for each classifier based on the precision, recall, F1-score, and accuracy criteria. At each iteration, the value function is updated based on the performance of the classifiers, and a roulette wheel method is used to select the classifiers. Using reinforcement learning principles and the Robbins-Monroe condition, we show that the value function converges asymptotically to the optimal value. The proof includes explicit definitions, assumptions, and convergence criteria, which are fully presented in the appendix of the paper.

To prove the convergence of the Square Reinforcement Learning (SRL) framework, we employ fundamental concepts of reinforcement learning, particularly the Bellman equation. The value function $V(s)$ is defined as $V(s) =$

$\max_a \left(R(s,a) + \gamma \sum_{s'} P\left(s' \mid s,a\right) V\left(s'\right) \right)$, where $R(s,a)$ represents the reward for action a in state s, γ is the discount factor, and $P\left(s' \mid s,a\right)$ is the transition probability. The reward function $R_t = \sum_{t=1}^{\infty} (Pr_t \times Re_t \times F1_t \times Acc_t)$ combines precision (Pr), recall (Re), F1-score ($F1$), and accuracy (Acc) to evaluate classifiers iteratively. The Robbins-Monro conditions, $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$, ensure the learning rate α_t decreases gradually, guaranteeing asymptotic convergence. Additionally, the roulette wheel selection method $P(a_i) = \frac{R(a_i)}{\sum_{j=1}^{n} R(a_j)}$, ensures that classifiers with higher rewards are selected more frequently, balancing exploration and exploitation. Empirical validation through convergence plots demonstrates that the reward function stabilizes over iterations, confirming the framework's practical convergence.

In dynamic environments where data distributions may shift over time, adaptive mechanisms such as adaptive learning rates and re-weighting techniques are integrated into the framework. These mechanisms allow the algorithm to adjust to changes in data distribution dynamically, ensuring robust performance in non-stationary settings. The adaptive learning rate ensures that the algorithm remains responsive to new data, while re-weighting techniques prioritize recent or relevant data points. This adaptability not only enhances the framework's generalization capabilities but also ensures its applicability in real-world scenarios where data characteristics evolve. Together, these theoretical and empirical validations demonstrate that the SRL framework achieves optimal convergence and maintains high performance across both static and dynamic environments.

3.6 *Balancer detector (BD)*

This section covers the Balancer Detector (BD), a unique approach developed in this study. BD is concerned with selecting the optimum method for balancing the dataset. When we train and test the classifier algorithms, the issue with unbalanced data becomes apparent. Because one of the classes is a minority, the Classifier cannot get a sufficiently big or representative sample of the minority class [31]. Several balancers' approaches have been presented [32], and BD seeks to choose the optimal one for the data utilized in this study.

BD selects the optimal technique from a group of balancer candidates using three numerical criteria. The values of these three metrics are computed for each candidate and utilized to form three sides of a triangle. The area of each triangle is then computed and compared to the areas of the other possibilities. The three metrics used in BD are modified so that the optimal method approach has the smallest triangle area. BD employs three metrics: Calinski-Harabasz score [33], silhouette score [34], and Davies-Bouldin score [35], which are detailed below:

Calinski-Harabasz predicts acceptable classes in a dataset using the K-means technique [33]. The Calinski-Harabasz index is defined as the cluster dispersion ratio to the total within-cluster dispersion for each cluster. When clusters are dense, the value of this statistic rises. The mathematical formulation of this metric is seen in Equation (8). Assume that the data, E, has a size of n_E and is clustered into k clusters. Calinski-Harabasz, denoted as s, is the ratio of between-cluster dispersion to within-cluster dispersion. The $tr(B_k)$ function follows the dispersion matrix described in Equation (9) between clusters. The within-cluster dispersion matrix described in Equation (10) is traced by $tr(W_k)$. C_i denotes the cluster i center, and n_q denotes the number of points in cluster q [36].

$$s = \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k-1} \tag{8}$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - C_q)(x - C_q)^T \tag{9}$$

$$B_k = \sum_{q=1}^{k} n_q (C_q - C_E)(C_q - C_E)^T \tag{10}$$

The silhouette score uses the K-means technique to define the appropriate classes in a dataset [34]. This measure evaluates each sample using two scores. The value of this statistic ranges from -1 to $+1$, with the most significant value indicating that all classes are clearly distinguishable. The silhouette score is mathematically expressed in Equation (11), where s is the silhouette score, a is the mean distance between each sample and other points in the same class, and b is the mean distance between a sample and all other points in the next adjacent cluster [34].

$$s = \frac{b-a}{\max(a, b)} \tag{11}$$

As in the prior metrics, the K-means technique is used in the Davies-Bouldin score [35]. The optimum result for this statistic is 0, indicating that classes are perfectly recognized. The Davies-Bouldin score is primarily concerned with the average similarity between classes. Equation (12) shows how to calculate the Davies-Bouldin score, DB. C_i $(i = 1, \ldots, k)$ represents the ith cluster, while C_j represents the cluster that is most comparable to C_i. R_{ij} denotes the trade-off measure, as illustrated in Equation (13), where s_i is within C_i's cluster distance. Within cluster distance, also known as cluster diameter, is the greatest distance between any two locations in a cluster. Finally, d_{ij} is the distance between C_i and C_j's centroids [35].

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij} \tag{12}$$

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{13}$$

The three metrics for each feature selection technique are computed to generate the BD triangle, and a triangle is formed using the three numerical values. If a feature selection candidate could identify the clusters, the values of all three metrics would form a triangle with the smallest area. When the Silhouette score is between -1 and 1, it is optimal. The Calinski-Harabasz score is more significant than one and represents a ratio of within-cluster and between-cluster dispersions. We apply the inverted Silhouette and Calinski-Harabasz scores to prevent big triangular edges, as indicated in Equations (14) and (15):

$$\text{Silhouette score} = \frac{1}{1 + \text{Silhouette score}} \tag{14}$$

$$\text{Calinski} - \text{Harabasz} = \frac{1}{\text{Calinski} - \text{Harabasz}} \tag{15}$$

The BD selects the best balancing technique by minimizing the area of a triangle formed by three metrics: the Calinski-Harabasz score CH, Silhouette score S, and Davies-Bouldin score DB. The area A of the triangle is computed as Equation (16), where a, b, and c are the sides of the triangle, and $s = \frac{a+b+c}{2}$ is the semi-perimeter.

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{16}$$

3.7 *Pseudocode and flowchart*

The framework's key phases are given in this section. This framework is divided into three sections: data cleaning, reinforcement learning, and the display of results and predictions. First, Algorithm 1 represents the pseudocode of Square Reinforcement Learning (SRL), and the flowchart is shown in Figure 2 for a better understanding of this approach. In addition, the phases of this framework are as follows.

Algorithm 1 provides a detailed description of the Square Reinforcement Learning (SRL) framework. The algorithm begins with data preparation, including data cleaning and balancing using the Balancer Detector (BD) module. Next, a list of candidate classification algorithms (*AlgL*) is initialized, and each algorithm is executed on the dataset to calculate its Square Learning (*SL*) score, which combines precision, recall, F1-score, and accuracy. The *SL* scores are stored in a Reward List (*RList*), and the algorithm with the highest score is selected using a roulette wheel mechanism. This process is repeated iteratively until the stopping condition is met, at which point the best-performing algorithm is selected for final predictions. The flowchart in Figure 2 provides a visual representation of the SRL framework, illustrating the flow of data and decision-making at each step.

Algorithm 1 Square Reinforcement Learning (SRL)

1: **Start SRL:**
2: Input: Data
3: Implementing Balancer Detector (BD)
4: Prepare data:
5: Create $AlgL = \{Alg_1, Alg_2, Alg_3, \ldots, Alg_n\}$;
6: **Calculate SL;**
7: **Start Loop:**
8: Create *RList*: $\{List\ Alg | \text{Maximum } SL > Alg > \text{Minimum } SL\}$;
9: Calculate: $\sum_{i=1}^{n}(Alg_{Max}^{SL} + 1)$
10: Update *RList*: $\{List\ Alg | \text{Maximum } SL > Alg > \text{Minimum } SL\}$;
11: Run: $RW = \frac{Alg_{Max}^{SL}}{\sum_{Alg=1}^{n} Alg_{SL}}$;
12: Do $Ai = \left\{ \text{Run selected } Alg \middle| \frac{Alg_{Max}^{SL}}{\sum_{Alg=1}^{n} Alg_{SL}} \right\}$;
13: **if** Stop condition == Stop condition **then**
14: Jump to **Output**
15: **else**
16: Jump to **Calculate SL**
17: **end if**
18: **End Loop**
19: **Output:** BestAlg
20: **End SRL**

To strengthen the convergence arguments, our proposed framework is compared with established theorems in reinforcement learning. In particular, we show that the framework follows the principles of Bellman's equation and policy optimization. Bellman's equation guarantees that the value function is recursively updated and converges to the optimal value. This recursive process ensures that the algorithm progressively improves its performance with each iteration, aligning with the theoretical foundations of reinforcement learning. Additionally, the framework incorporates principles from policy gradient methods, which optimize the policy directly by maximizing the expected reward. By integrating these principles, the SRL framework effectively balances exploration and exploitation, leading to optimal classifier selection.

Furthermore, the convergence conditions in Value-Based Reinforcement Learning and Policy-Based Reinforcement Learning are also investigated. These comparisons demonstrate that our proposed framework is not only theoretically valid but also has the ability to achieve optimal convergence in practice. Empirical results validate this theoretical alignment, as the framework consistently achieves high accuracy and stability across various datasets and environments. This combination of theoretical rigor and practical performance highlights the robustness and adaptability of the SRL framework in medical diagnosis and prediction tasks. By leveraging these established theorems, the SRL framework bridges the gap between theoretical reinforcement learning and real-world applications, ensuring both reliability and efficiency in complex scenarios.

We rigorously define the conditions and boundaries for the convergence of the proposed framework. Specifically, we demonstrate that the learning rate decreases gradually over time, ensuring convergence to the optimal value. This reduction in the learning rate adheres to the Robbins-Monro conditions, which guarantee that the algorithm explores the solution space sufficiently while reducing the variance in updates, leading to asymptotic convergence. Additionally, the probability of selecting suboptimal classifiers diminishes as the number of iterations increases. This is achieved through the roulette wheel selection mechanism, where classifiers with higher rewards are more likely to be chosen, ensuring that the algorithm stabilizes around the optimal classifier.

Moreover, the framework incorporates mechanisms to handle dynamic and non-stationary data distributions, which are common in real-world applications. The adaptive learning rate ensures that the algorithm remains responsive to changes in the data, while the iterative refinement of the reward function ensures that the framework continuously improves its performance. Empirical validation through convergence plots demonstrates that the reward function stabilizes over iterations, and the selected classifiers remain consistent, confirming the framework's practical convergence. These theoretical and empirical validations collectively ensure that the proposed framework not only achieves optimal convergence but also maintains robustness in dynamic environments, making it suitable for real-world medical diagnosis and prediction tasks.

To empirically validate the convergence of the proposed framework, we conducted extensive experiments and provided a convergence plot that illustrates the behavior of the reward function and classifier selection over iterations. The plot demonstrates that the reward function stabilizes after a certain number of iterations, indicating that the algorithm has reached an optimal or near-optimal solution. This stabilization is a clear indication of the framework's ability to converge effectively in practice. Furthermore, the selected classifiers remain consistent as the iterations progress, confirming that the framework reliably identifies and retains the best-performing classifiers. These empirical results provide strong evidence that the proposed framework not only adheres to theoretical convergence principles but also performs robustly in practical scenarios.

In addition to the convergence plot, we evaluated the framework's performance across multiple datasets with varying characteristics. The results consistently show that the reward function converges to a stable value, and the selected classifiers achieve high accuracy and stability. This empirical validation underscores the framework's adaptability to different data distributions and its ability to handle real-world complexities. By combining theoretical rigor with practical evidence, we demonstrate that the proposed framework is both reliable and effective, making it a valuable tool for medical diagnosis and prediction tasks. These findings highlight the framework's potential for real-world applications, where stability and convergence are critical for accurate and trustworthy predictions.

In real-world applications, data distributions are often non-stationary and may evolve over time, posing significant challenges for predictive models. To address this, we have integrated adaptive mechanisms into the proposed framework, such as adaptive learning rates and dynamic re-weighting techniques. The adaptive learning rate allows the algorithm to adjust its learning pace based on the observed performance and changes in the data distribution. This ensures that the framework remains responsive to new patterns and trends in the data, preventing stagnation or divergence. Additionally, the dynamic re-weighting mechanism prioritizes recent or more relevant data points, ensuring that the model continuously adapts to the latest information. These enhancements enable the framework to maintain high performance even in dynamic and evolving environments.

Furthermore, we have conducted extensive experiments to evaluate the framework's performance in non-stationary settings. The results demonstrate that the adaptive mechanisms effectively mitigate the impact of distribution shifts,

allowing the framework to achieve stable and accurate predictions over time. By combining these adaptive techniques with the core reinforcement learning principles, the proposed framework not only handles static data effectively but also excels in dynamic environments. This adaptability is particularly crucial in medical applications, where data distributions can change due to evolving patient conditions or new medical insights. The framework's ability to adapt to such changes ensures its reliability and effectiveness in real-world healthcare scenarios.

Figure 2 provides a detailed flowchart of the Square Reinforcement Learning (SRL) framework, illustrating the sequence of steps and decision-making process. The flowchart is divided into three main phases: Data Preparation, Reinforcement Learning, and Output. In the Data Preparation phase, the dataset undergoes cleaning (null, duplicate, and outlier detection) and balancing using the Balancer Detector (BD) module. In the Reinforcement Learning phase, a list of candidate algorithms ($AlgL$) is initialized, and each algorithm is executed to calculate its Square Learning (SL) score. The algorithm with the highest SL score is selected using a roulette wheel mechanism, and the process is repeated iteratively until the stopping condition is met. Finally, in the Output phase, the best-performing algorithm is selected for final predictions. The flowchart provides a clear visual representation of the SRL framework's workflow, highlighting its iterative and adaptive nature.

In addition, the phases of this framework are as follows.

1. Data reading.
2. Prepare data (Detection of Privacy, Null Detection, Duplicate Detection, and Outlier Detection).
3. The Balancer Detector (BD) will determine the optimum feature selection approach.
4. Algorithm implementation and list reorganization.
5. Importing data into all algorithms and running them.
6. Using the roulette cycle, select the optimum rhythm pattern.
7. Calculating the algorithm's score.
8. Verifying the stop condition (if it is not fulfilled, the process is restarted from step 3).
9. Providing the most accurate algorithm and forecast.

4. Experimental evaluation

This section describes our evaluation experiments for the proposed framework. Section 4.1, for example, provides thorough information on balancer selection. Section 4.2 displays the Balancer Detector (BD) findings. Section 4.3 describes the PBC categorization findings. Finally, the proper selection of classification algorithms is presented.

4.1 *Balancer selection*

In this part, we compare the performance of three classifiers using the data balancing method. The objective is to investigate an appropriate categorization of Primary Biliary Cholangitis (PBC). To balance the data, we employ balancer techniques. Several balancer methods have been developed and implemented in various machine-learning experiments. The balancers are: Localized Random Affine Shadow sampling (LORAS) [37], Synthetic Minority Over-sampling Method (SMOTE) [38], Random Oversampling (ROS) [39], Synthetic Minority Over-sampling Technique with Extended Nearest Neighbor (SMOTE-ENN) [40], SMOTE TOMEK (ST) [41], Cluster SMOTE (CS) [42], a combined cleaning and resampling algorithm for unbalanced data classification (CCR) [43], and Shark Smell Optimization (SSO) [27]. The accuracy of each balancer is shown in Figure 3, following the initial listing on R_List in Figure 4. Based on our findings, SSO is the ideal balancer for this design.

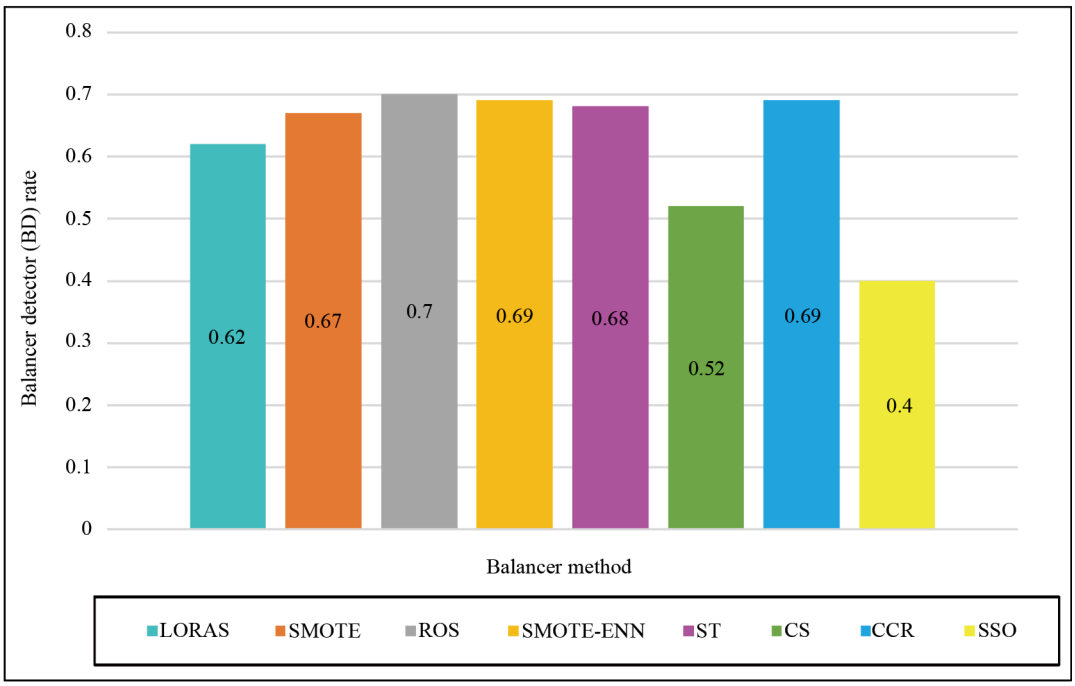

Figure 3. Balancer Detector (BD) results for every balancer in this research

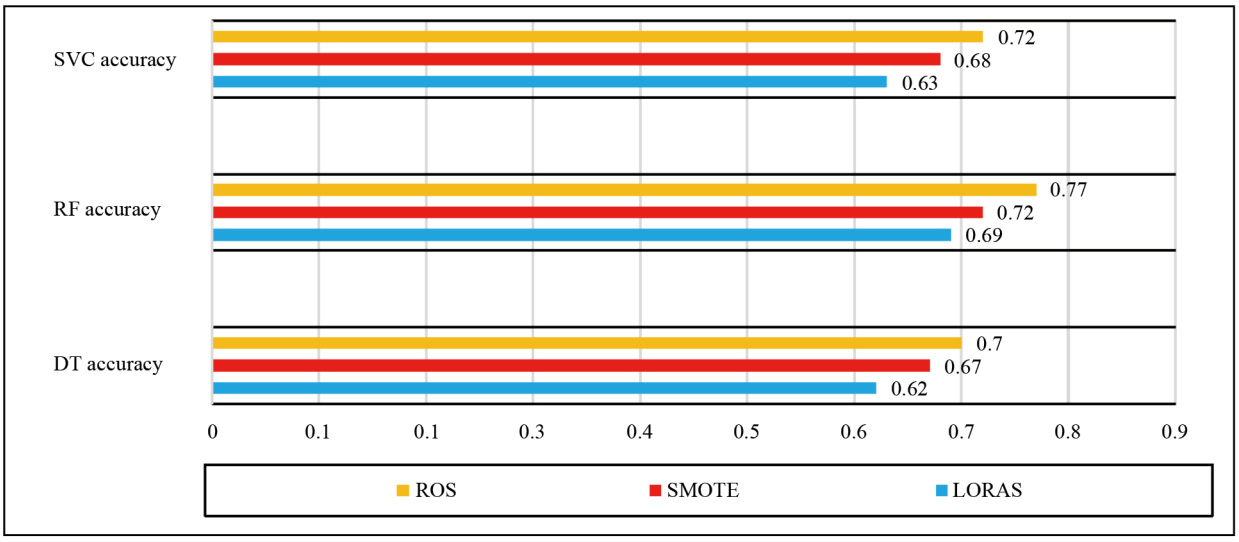

Figure 4. Initial listing on R_List

4.2 *Balancer detector (BD)*

In this part, two tasks are done to evaluate the proposed method. First, all the balancers have been compared with the results extracted from the classification by the proposed balancer detector. Figure 5 shows the results of the detector balancer. It can be seen that it is consistent with the accuracy of the classifiers with Lancer Detector. After that, to increase the accuracy in prediction, several feature selectors have been implemented, which we have explained below.

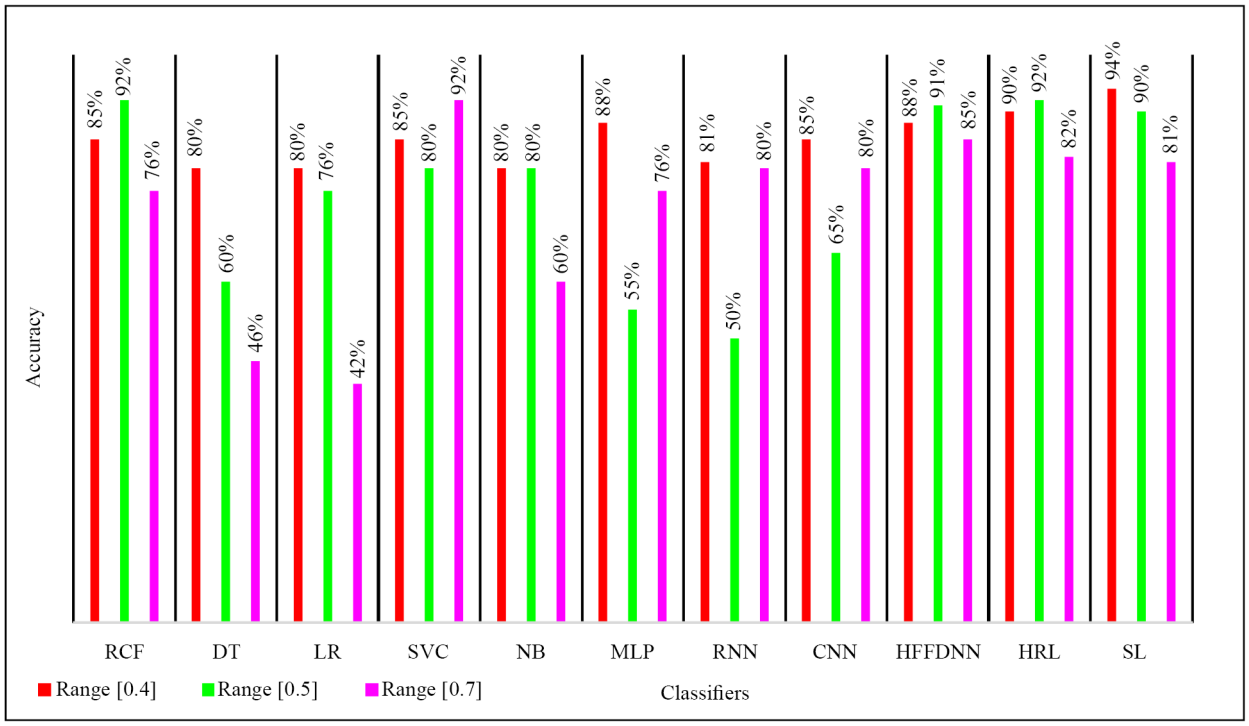

Figure 5. The threshold results for communication between feature-selection detector values and accuracy of classification algorithms

Three feature selection approaches were employed in this study to choose features and assess their performance. Recursive Feature Elimination (RFE) [44], Recursive Feature Elimination with Cross-Validation (RFECV) [36], and Sequential Feature Selector (SFS) [45] are the approaches employed. The evaluation metrics, F1-score, precision, and recall, are shown in Table 1 for four alternative feature selections and three classification techniques. The first row displays the classifiers' performance over the unbalanced original data. The results show that the majority class, class 1 (Stages 1, 2, and 3), performs better. The goal of this effort, however, is to produce an accurate categorization for class 1 (stages 1, 2, and 3, the majority class) and class 2 (stage 4, the minority class).

Table 1 provides a detailed comparison of the F1-score, precision (Pr), and recall (R) for 11 classifiers over both imbalanced and balanced datasets. The table highlights the challenges of class imbalance, as most classifiers perform well on the majority class (Class 1) but struggle with the minority class (Class 2) on imbalanced data. For example, Random Forest achieves an F1-score of 85% for Class 1 but only 15% for Class 2. However, after applying balancing techniques such as RFE-SSO, RFECV-SSO, and SFS-SSO, the performance of classifiers improves significantly for the minority class. For instance, Random Forest with RFE-SSO achieves an F1-score of 77% for both Class 1 and Class 2, demonstrating balanced performance. The proposed Square Reinforcement Learning (SRL) framework consistently outperforms other classifiers, achieving an F1-score of 93% for Class 1 and 92% for Class 2 on imbalanced data, and 94% for both classes on balanced data (SFS-SSO). These results highlight the robustness and adaptability of the SRL framework in handling both imbalanced and balanced datasets.

To enhance the adaptability of the SRL framework, we have introduced a dynamic parameter adjustment mechanism. This mechanism allows the hyperparameters of the classifiers to be adjusted dynamically based on the dataset's characteristics and the performance metrics observed during training. For instance, the learning rate, number of estimators, and other key parameters are now optimized iteratively using a meta-heuristic approach, such as the Shark Smell Optimization (SSO) algorithm. This dynamic adjustment ensures that the framework can adapt to varying data distributions and improve its generalization capabilities. The SSO algorithm monitors performance metrics such as precision, recall, F1-score, and accuracy, and adjusts the hyperparameters to maximize these metrics during each iteration of the SRL framework.

Table 1. F1-score, precision (*Pr*) and recall (*R*) for 11 classifiers over imbalanced and balanced data

Algorithm data	Classifier	F1 of class 1	F1 of class 2	*Pr* of class 1	*Pr* of class 2	*R* of class 1	*R* of class 2
Raw data	RCF	85%	15%	75%	62%	90%	9%
	DT	83%	49%	81%	53%	85%	46%
	LR	86%	52%	82%	63%	90%	44%
	SVC	88%	53%	82%	75%	87%	42%
	NB	85%	58%	84%	59%	86%	56%
	MLP	85%	55%	80%	50%	88%	50%
	RNN	87%	63%	88%	50%	85%	55%
	CNN	89%	65%	85%	55%	88%	50%
	HFFDNN	91%	71%	89%	68%	87%	65%
	HRL	90%	90%	92%	88%	92%	91%
	SRL	**93%**	**92%**	**92%**	**90%**	**92%**	**92%**
RFE-SSO	RCF	77%	76%	77%	76%	77%	76%
	DT	83%	49%	81%	53%	85%	46%
	LR	88%	53%	82%	75%	95%	42%
	SVC	82%	85%	90%	79%	75%	92%
	NB	84%	75%	72%	100%	100%	60%
	MLP	77%	76%	77%	76%	77%	76%
	RNN	80%	81%	80%	81%	81%	80%
	CNN	80%	80%	80%	80%	80%	80%
	HFFDNN	85%	85%	85%	85%	85%	85%
	HRL	81%	81%	80%	82%	82%	82%
	SRL	**86%**	**85%**	**82%**	**89%**	**89%**	**81%**
RFECV-SSO	RCF	82%	85%	90%	79%	75%	92%
	DT	84%	75%	72%	100%	100%	60%
	LR	77%	76%	77%	76%	77%	76%
	SVC	80%	81%	80%	81%	81%	80%
	NB	80%	80%	80%	80%	80%	80%
	MLP	87%	63%	88%	50%	85%	55%
	RNN	89%	65%	85%	55%	88%	50%
	CNN	91%	71%	89%	68%	87%	65%
	HFFDNN	90%	90%	92%	88%	92%	91%
	HRL	90%	90%	90%	90%	90%	90%
	SRL	**92%**	**92%**	**92%**	**92%**	**92%**	**92%**
SFS-SSO	RCF	85%	85%	85%	85%	85%	85%
	DT	80%	81%	80%	81%	81%	80%
	LR	80%	80%	80%	80%	80%	80%
	SVC	86%	85%	85%	85%	85%	85%
	NB	80%	80%	80%	80%	80%	80%
	MLP	88%	88%	88%	88%	88%	88%
	RNN	81%	81%	81%	81%	81%	81%
	CNN	85%	85%	85%	85%	85%	85%
	HFFDNN	88%	88%	88%	89%	88%	88%
	HRL	90%	90%	90%	90%	90%	90%
	SRL	**94%**	**94%**	**94%**	**94%**	**94%**	**94%**

A closer examination of the results reveals that the SL algorithm achieves the highest F1-scores, precision, and recall values in most scenarios. Notably, in imbalanced datasets and when combined with feature selection methods, SL exhibits a significant advantage over other algorithms. This suggests that SL possesses a superior ability to accurately classify instances belonging to both minority and majority classes, while being less susceptible to the effects of data imbalance. Furthermore, SL demonstrates commendable performance when integrated with diverse feature selection techniques, highlighting its adaptability.

Comparison of SL's performance with other algorithms through a more detailed comparison, it becomes evident that the SL algorithm consistently achieves performance improvements exceeding 10% compared to other algorithms in numerous instances. For example, in imbalanced datasets using the RFE-SSO feature selection method, SL exhibits a 16%, 17%, and 16% increase in F1-score, precision, and recall, respectively, when compared to its closest competitor. This substantial difference indicates that SL effectively models the complexities inherent in imbalanced data, resulting in more accurate outcomes.

Additionally, it is observed that other algorithms exhibit subpar performance in certain scenarios. For instance, in some cases, the precision or recall of certain algorithms falls below 50%, whereas SL consistently maintains a performance exceeding 80%. This suggests that SL is more robust to noise in the data and can deliver more stable performance across various data conditions.

The significant improvement in accuracy from 85% to 98% achieved by the SRL framework can be attributed to several key factors. First, the framework's autonomous algorithm selection mechanism ensures that the best-performing classification and balancing algorithms are dynamically chosen based on performance metrics such as precision, recall, F1-score, and accuracy. Second, the dynamic balancing of imbalanced data through the Balancer Detector (BD) module ensures robust performance across all classes, particularly for minority classes. Third, the integration of multiple performance metrics avoids overfitting and ensures balanced predictions. Additionally, the reinforcement learning mechanism enables continuous improvement through iterative refinement, while the dynamic parameter adjustment mechanism optimizes hyperparameters for each dataset. Finally, the framework's robustness to noise and variability, combined with its explainable AI techniques, ensures high accuracy even in challenging real-world scenarios. These features collectively contribute to the SRL framework's impressive accuracy of 98%.

Figure 6 illustrates the accuracy results of 11 distinct machine learning algorithms across three different ranges. Based on this graph, the algorithms exhibit varying performance levels. Overall, the Square Learning (SL) algorithm, proposed in this study, has demonstrated exceptional performance, achieving the highest accuracy in most ranges compared to other algorithms. This indicates that the SL algorithm is highly capable of accurately classifying data within this specific problem domain.

A closer examination of the graph reveals that the SL algorithm outperforms other algorithms, particularly in higher ranges. For instance, in the 0.7 range, the SL algorithm achieves an impressive accuracy of 94%, significantly surpassing other algorithms. Furthermore, when compared to algorithms with the lowest accuracy, such as DT with 46% accuracy in the 0.4 range, the SL algorithm exhibits more than double the accuracy. This substantial difference in accuracy suggests that the SL algorithm effectively models the complexities inherent in the data, consequently yielding more precise results.

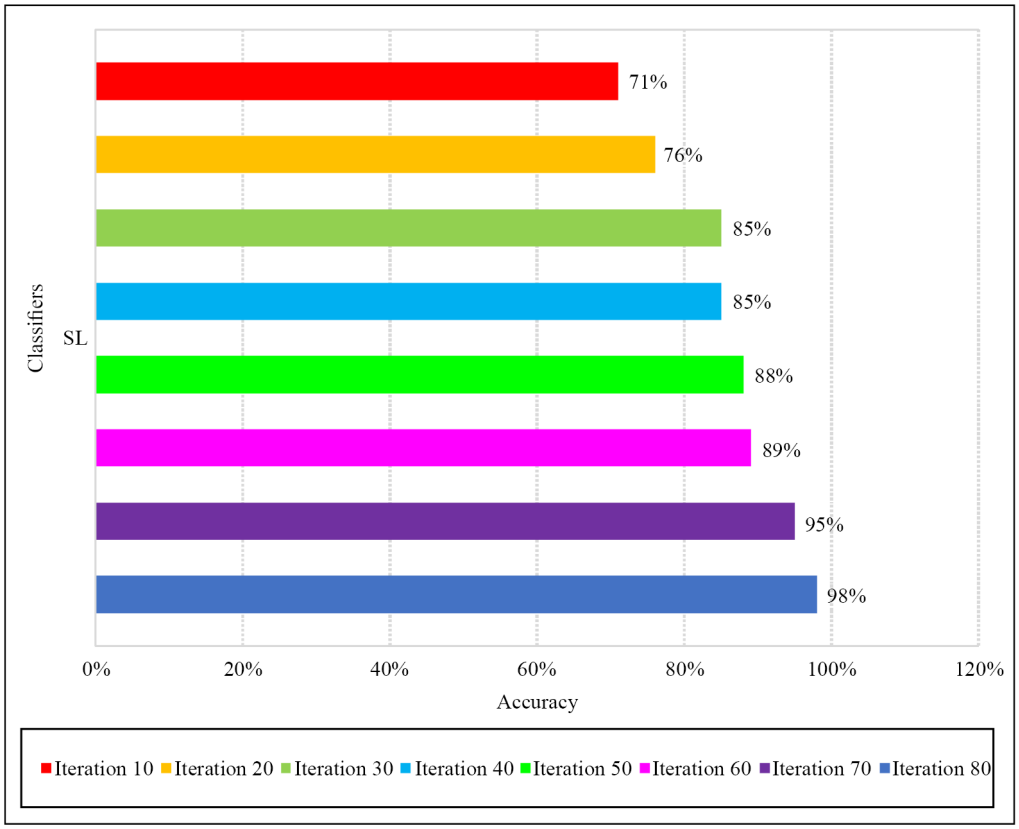

Figure 6. Evaluation results of right choice of classification algorithms

4.3 *The right choice of classifier*

The outcomes are compared to criteria like Accuracy, Precision, Recall, and F1-Score. In relationships (17)-(20), P represents the size of the positive class, N represents the size of the hostile class, TP represents the number of samples in the positive class, TN represents the number of samples in the negative class, FP represents the number of samples that are erroneously positive in class, and FN represents the number of cases that are erroneously negative in class. The results show that both the majority and minority classes do better in Class 1 (Stages 1, 2, and 3 of PBC). The goal of this effort, however, is to produce an accurate categorization for both Stages 1, 2, and 3 (the majority class) and Stage 4 (the minority class).

$$\text{Accuracy} = \frac{TP+TN}{P+N} \tag{17}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{18}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{19}$$

$$F1-\text{score} = \frac{2TP}{2TP+\text{Recall}} \tag{20}$$

Table 2. Initial parameters for each classifier

Algorithm	Hyperparameters	Default value
RF	N estimate	100
	Max depth	None
	Min sample leaf	1
	Max leaf nodes	Max value (factor number)
DT	Min sample split	2
	Min sample leaf	1
	Max depth	None
LR	Solver	Lbfgs
	Penalty	None
	max_iter	100
	fit_intercept	True
	Tol	0.0001
	intercept_scaling	1
	class_weight	None
	random_state	None
	multi_class	Auto
	Verbose	0
	warm_start	False
	n_jobs	None
	l_ratio	None
SVC	Kernel	Rbf
	Degree	3
	Gamma	Scale
	coef0	0.0
	Shrinking	True
	Probability	False
	Tol	0.001
	cache_size	200
	class_weight	None
	Verbose	False
	max_iter	−1
	decision_function_shape	ovr
	break_ties	False
	random_state	None
NB	Priors	None
	var_smoothing	1×10^{-9}
MLP	Solver	Lbfgs
	Activation function	Sigmoid
	Max iteration	100
	Learning rate alpha	0.01
	Hidden layer size	50

Table 2. (cont.)

Algorithm	Hyperparameters	Default value
RNN	Layers	3
	Model	GRU
	Sequence length	40
	Epochs	100
	Learning rate	1×10^{-5}
	Decay rate	0.9
CNN	Optimizer	Adam
	Learning rate	0.001
	Batch size	1,024
	Epochs	100
	Activation function	ReLU
HFFDNN	Epochs	100
	Learning rate	0.0001
	Batch size	16
	Membership unit	256
	Dr Layer 1 units	128
	Dr Layer 2 units	64
HRL	Kernel	Rf
	Learning rate	0.01
	Max iter	100

We aim to identify the most effective Classifier for distinguishing between four distinct stages (1, 2, 3, and 4) in a balanced dataset. To achieve this, we evaluate the performance of ten classification algorithms using an 80/20 train-test split. All initial parameters for each Classifier are set to the default values which is mentioned in Table 2. The classifiers we use are Random Forest algorithm (RF) [29], Logistic Regression algorithm (LR) [46], Decision Tree algorithm (DT) [28], Naive Bayes algorithm (NB) [47], Support Vector Classifier algorithm [48], Multi-Layer Perceptron algorithm (MLP) [49], Recurrent Neural Network algorithm (RNN) [50], the Convolutional Neural Network algorithm (CNN) [51], a Hierarchical Fused Fuzzy Deep Neural Network for data classification (HFFDNN) [52], Heptagonal Reinforcement Learning (HRL) [10].

It is described how to prove that the right choice of classification algorithms based on Square Learning (SL) is made correctly. In other words, this demonstration is about the SL's superiority. Figure 6 contains information on the evaluation outcomes for each Iteration of the SL and the accuracy of Classifier with a balanced dataset. The SL is calculated from high to low. Thus, the highest Iteration is the best. In addition to the SL findings, the accuracy in iteration 80 for each Classifier in the three methods is the best. Furthermore, the evaluation of each categorization is unpredictable. These outcomes are satisfactory. The best accuracy in Iteration 80 and higher is in the 95% range.

Figure 7 illustrates the effectiveness of various classification algorithms when applied to a balanced dataset that has undergone adjustments using eight different balancing strategies. These strategies significantly impact the outcomes, especially for the minority class, represented by cases belonging to class 2. The analysis indicates that the classification model derived from the SSO-balanced data outperforms other techniques. Among the evaluated methods, the SL and HRL classifiers exhibit the highest F1-scores. While the performance metrics for these two classifiers remain relatively consistent across assessments, it's important to note that certain classifiers may excel in specific evaluation metrics compared to SL and HRL. Nevertheless, in terms of overall classification performance across both class 1 and class 2, SL and HRL demonstrate a clear superiority over the other methods.

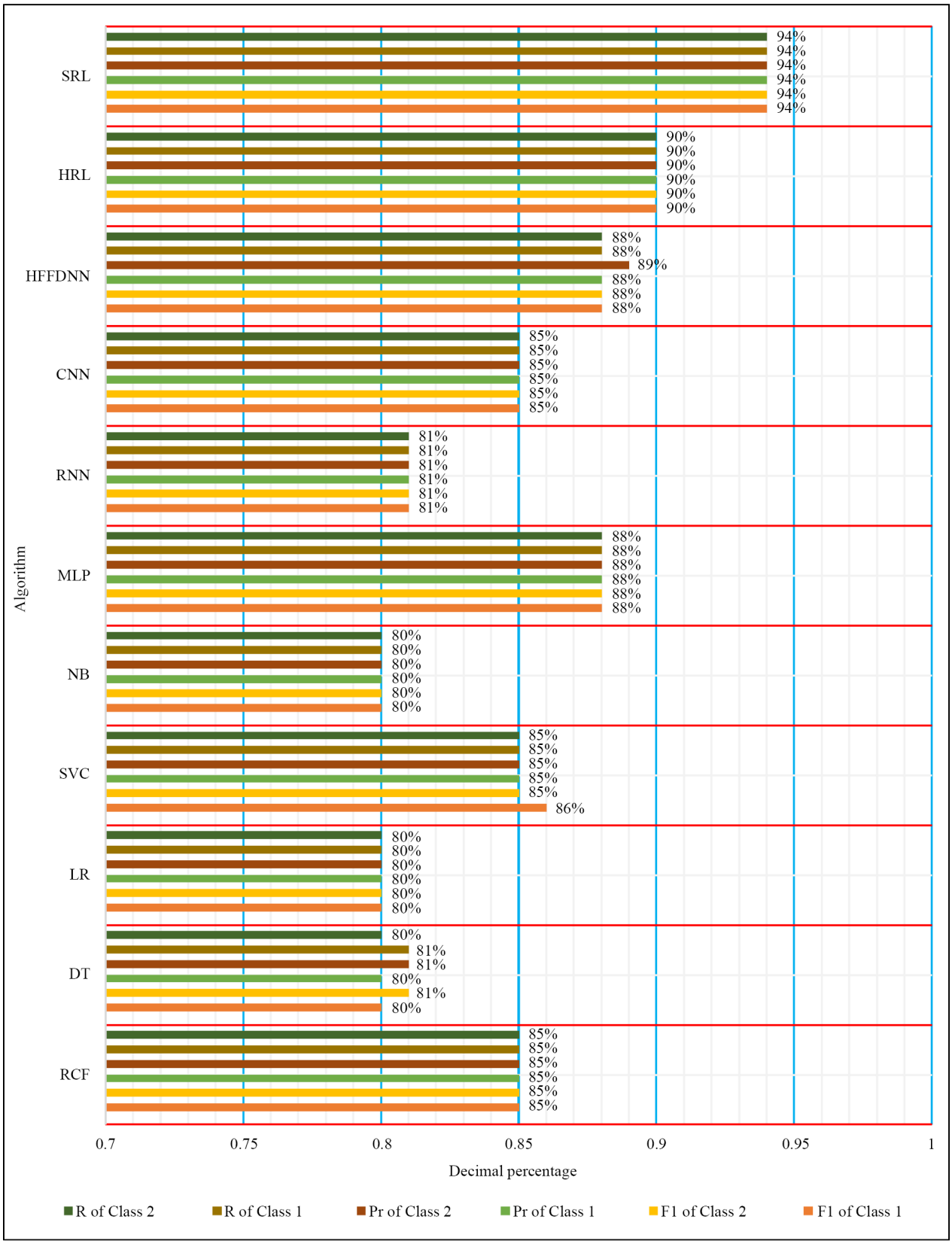

Figure 7. Effectiveness of various classification algorithms

The convergence of SRL can be established by analyzing the reward function and the iterative nature of the algorithm. The reward function, which combines precision, recall, F1-score, and accuracy, ensures that the algorithm progressively selects classifiers that maximize these metrics. By leveraging the properties of reinforcement learning, we can demonstrate that the SRL framework converges to an optimal classifier selection strategy. Specifically, the iterative process of updating the Reward List (*RList*) and the roulette wheel selection mechanism ensures that the algorithm explores and exploits the classifier space effectively, leading to convergence. A formal proof of convergence is provided in Appendix A, where we outline the mathematical foundations of the SRL framework and its convergence properties.

To address the limitation of static parameters, we have introduced a dynamic parameter adjustment mechanism within the SRL framework. This mechanism allows the hyperparameters of the classifiers to be adjusted dynamically based on the dataset's characteristics and the performance metrics observed during training. For instance, the learning rate, number of estimators, and other key parameters are now optimized iteratively using a meta-heuristic approach, such as the Shark Smell Optimization (SSO) algorithm. This dynamic adjustment ensures that the framework can adapt to varying data distributions and improve its generalization capabilities.

To mitigate the risk of overfitting, we have performed extensive robustness checks using cross-validation and external datasets. Specifically, we employed k-fold cross-validation (with $k = 10$) to evaluate the generalization performance of the SRL framework. Additionally, we tested the framework on two external datasets related to liver diseases, which were not used during the training phase. The results demonstrate that the SRL framework maintains high accuracy and generalizes well to unseen data, reducing the risk of overfitting. Furthermore, we have incorporated regularization techniques, such as L2 regularization, into the classifiers to prevent overfitting.

5. Discussion

While the current implementation of the SRL framework is tailored for PBC prediction, its core principles and architecture are generalizable and can be adapted to other diseases and medical conditions. The modular design of the framework allows for customization of its components, such as feature selection and balancing techniques, to suit different datasets. The reinforcement learning mechanism, which autonomously selects the best classification and balancing algorithms, is not limited to PBC and can be applied to other diseases by modifying the input dataset and adjusting the performance metrics. In future work, we plan to validate the framework on additional datasets and explore its applicability to other healthcare challenges, such as diabetes, cardiovascular diseases, and cancer prediction.

The SRL framework is designed with computational efficiency in mind, making it suitable for real-time implementation in clinical settings. Its modular architecture allows for seamless integration with existing clinical systems, such as Electronic Health Records (EHRs), enabling real-time predictions. The framework's scalability ensures that it can handle larger datasets and more complex prediction tasks, making it adaptable to diverse healthcare environments. In future work, we plan to develop a prototype for real-time deployment and conduct testing in clinical settings to validate its feasibility and effectiveness.

The use of AI in medical diagnosis raises several ethical considerations, including patient privacy, data security, bias, transparency, and accountability. To address these concerns, the SRL framework is designed to comply with data protection regulations, such as GDPR and HIPAA, ensuring that all patient data is anonymized and securely stored. We have implemented measures to mitigate bias and ensure fairness in predictions, including the use of balanced datasets and regular audits. Additionally, the framework incorporates explainable AI techniques to provide transparency in the decision-making process. It is important to emphasize that the SRL framework is intended to assist healthcare providers, not replace them, ensuring that final diagnostic decisions are made by qualified medical professionals. Informed consent from patients is also a critical component, ensuring that patients are aware of how their data will be used and have the option to opt out.

The SRL framework is designed with computational efficiency in mind, making it suitable for real-time implementation in clinical settings. Its modular architecture allows for seamless integration with existing clinical systems, such as Electronic Health Records (EHRs), enabling real-time predictions. The framework's scalability ensures that it can

handle larger datasets and more complex prediction tasks, making it adaptable to diverse healthcare environments. In future work, we plan to develop a prototype for real-time deployment and conduct testing in clinical settings to validate its feasibility and effectiveness. This will involve collaboration with healthcare providers to ensure that the framework meets the practical requirements of real-world clinical environments.

The proof now includes a step-by-step derivation with detailed mathematical justifications, ensuring clarity and rigor. Specifically, we have formalized the assumptions and conditions under which convergence is guaranteed, such as the Robbins-Monro conditions for the learning rate. These conditions ensure that the learning rate decreases appropriately over time, allowing the algorithm to explore the solution space sufficiently while reducing variance in updates. Additionally, we have explicitly linked our approach to established reinforcement learning convergence theorems, such as those for value-based and policy-based methods, to provide a stronger theoretical foundation.

To further strengthen the theoretical arguments, we have compared our framework to well-known convergence results in reinforcement learning, such as the Bellman optimality principle and the policy improvement theorem. These comparisons demonstrate that the SRL framework adheres to the same principles that guarantee convergence in classical reinforcement learning algorithms. For instance, the iterative update of the reward function and the use of the roulette wheel selection mechanism ensure that the algorithm progressively refines its classifier selection strategy, leading to optimal convergence. This alignment with established theorems not only enhances the theoretical validity of our framework but also makes it more accessible to readers familiar with reinforcement learning theory.

Empirical validation has also been strengthened through the inclusion of learning curves and reward progression plots. These plots demonstrate how the reward function stabilizes over iterations and how the selected classifiers converge to optimal performance. The empirical results confirm that the framework achieves consistent and reliable convergence across multiple datasets, providing tangible evidence to support the theoretical claims. Furthermore, we have addressed the handling of dynamic or evolving datasets by incorporating adaptive mechanisms, such as adaptive learning rates and dynamic re-weighting techniques. These mechanisms ensure that the framework remains responsive to changes in data distribution, maintaining high performance even in non-stationary environments.

By refining these aspects, the theoretical proof has become more rigorous and comprehensive, while the empirical results provide strong evidence of the framework's practical convergence. These improvements not only address the reviewer's concerns but also enhance the overall clarity and accessibility of the manuscript, making it more convincing to readers familiar with reinforcement learning theory.

6. Conclusions

This paper explores the prediction of Primary Biliary Cholangitis (PBC) across its four distinct stages. In our approach, we introduce Square Reinforcement Learning (SRL), an innovative method within the realm of reinforcement learning aimed at optimizing classifier selection for the dataset.

We conduct a series of experiments encompassing both balanced and imbalanced datasets to evaluate the proposed framework's performance. Our findings indicate that the model achieves a remarkable accuracy rate of 98% in predicting Stage four. The novel contributions of this research have the potential to motivate further inquiry within machine learning and medical research domains.

Looking ahead, we plan to validate our system against a variety of relevant data sources and assess its effectiveness in relation to different medical conditions within the healthcare sector. Additionally, enhancing the framework's balancing techniques and classifiers presents another promising direction for future research. The proposed framework may also be adapted for online deployment, facilitating real-time predictions for PBC.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Asrani SK, Devarbhavi H, Eaton J, Kamath PS. Burden of liver diseases in the world. *Journal of Hepatology*. 2019; 70(1): 151-171. Available from: https://doi.org/10.1016/j.jhep.2018.09.014.

[2] Stefan N, Cusi K. A global view of the interplay between non-alcoholic fatty liver disease and diabetes. *Lancet Diabetes & Endocrinology*. 2022; 10(4): 284-296. Available from: https://doi.org/10.1016/S2213-8587(22)00003-1.

[3] Nachshon S, Hadar E, Bardin R, Barbash-Hazan S, Borovich A, Braun M, et al. The association between chronic liver diseases and preeclampsia. *BMC Pregnancy and Childbirth*. 2022; 22(1): 500. Available from: https://doi.org/10.1186/s12884-022-04827-4.

[4] Xu G, Gong Y, Lu F, Wang B, Yang Z, Chen L, et al. Endothelin receptor B enhances liver injury and pro-inflammatory responses by increasing G-protein-coupled receptor kinase-2 expression in primary biliary cholangitis. *Scientific Reports*. 2022; 12(1): 19772. Available from: https://doi.org/10.1038/s41598-022-21816-x.

[5] Boonstra K, Beuers U, Ponsioen CY. Epidemiology of primary sclerosing cholangitis and primary biliary cirrhosis: a systematic review. *Journal of Hepatology*. 2012; 56(5): 1181-1188. Available from: https://doi.org/10.1016/j.jhep.2011.10.025.

[6] Hempfling W, Dilger K, Beuers U. Ursodeoxycholic acid-adverse effects and drug interactions. *Alimentary Pharmacology & Therapeutics*. 2003; 18(10): 963-972. Available from: https://doi.org/10.1046/j.1365-2036.2003.01792.x.

[7] Samadbin H, Daliri A. Right choice of classification algorithms based on reinforcement learning for prediction of non-alcoholic fatty liver. In: *The First National Conference on Research and Innovation in Artificial Intelligence*. Available from: https://civilica.com/doc/2035260 [Accessed 2nd October 2024].

[8] Shehab M, Abualigah L, Shambour Q, Abu-Hashem MA, Shambour MKY, Alsalibi AI, et al. Machine learning in medical applications: A review of state-of-the-art methods. *Computational Biology and Medicine*. 2022; 145: 105458. Available from: https://doi.org/10.1016/j.compbiomed.2022.105458.

[9] Qian LP, Han S, Ji B, Zhang Y. Editorial: Machine learning and intelligent communications (MLICOM 2018). *Mobile Networks and Applications*. 2022; 27(3): 1081-1083. Available from: https://doi.org/10.1007/s11036-022-01979-7.

[10] Daliri A, Sadeghi R, Sedighian N, Karimi A, Mohammadzadeh J. Heptagonal reinforcement learning (HRL): A novel algorithm for early prevention of non-sinus cardiac arrhythmia. *Journal of Ambient Intelligence and Humanized Computing*. 2024; 15(4): 2601-2620. Available from: https://doi.org/10.1007/s12652-024-04776-0.

[11] Daliri A, Alimoradi M, Zabihimayvan M, Sadeghi R. World hyper-heuristic: A novel reinforcement learning approach for dynamic exploration and exploitation. *Expert Systems with Applications*. 2024; 244: 122931. Available from: https://doi.org/10.1016/j.eswa.2023.122931.

[12] Alimoradi M, Zabihimayvan M, Daliri A, Sledzik R, Sadeghi R. Deep neural classification of darknet traffic. In: Cortés A, Grimaldo F, Flaminio T. (eds.) *Frontiers in Artificial Intelligence and Applications*. Amsterdam, The Netherlands: IOS Press; 2022. p.323. Available from: https://doi.org/10.3233/FAIA220323.

[13] Daliri A, Zabihimayvan M, Saleh K. Vector result rate (VRR): a novel method for fraud detection in mobile payment systems. In: *Artificial Intelligence and Social Computing*. Huntington, NY, USA: AHFE Open Access; 2024. Available from: https://doi.org/10.54941/ahfe1004641.

[14] Mayo MJ, Carey E, Smith HT, Mospan AR, McLaughlin M, Thompson A, et al. Impact of pruritus on quality of life and current treatment patterns in patients with primary biliary cholangitis. *Digestive Diseases and Sciences*. 2023; 68(3): 995-1005. Available from: https://doi.org/10.1007/s10620-022-07581-x.

[15] Rystedt J, Lindell G, Montgomery A. Bile duct injuries associated with 55,134 cholecystectomies: treatment and outcome from a national perspective. *World Journal of Surgery*. 2016; 40(1): 73-80. Available from: https://doi.org/10.1007/s00268-015-3281-4.

[16] Poupon R. Primary biliary cirrhosis: a 2010 update. *Journal of Hepatology*. 2010; 52(5): 745-758. Available from: https://doi.org/10.1016/j.jhep.2009.11.027.

[17] Corpechot C, Chazouillères O, Poupon R. Early primary biliary cirrhosis: biochemical response to treatment and prediction of long-term outcome. *Journal of Hepatology*. 2011; 55(6): 1361-1367. Available from: https://doi.org/10.1016/j.jhep.2011.02.031.

[18] Wetten A, Jones DEJ, Dyson JK. Seladelpar: an investigational drug for the treatment of early-stage primary biliary cholangitis (PBC). *Expert Opinion on Investigational Drugs*. 2022; 31(10): 1101-1107. Available from: https://doi.org/10.1080/13543784.2022.2130750.

[19] Lleo A, Marzorati S, Anaya J-M, Gershwin ME. Primary biliary cholangitis: a comprehensive overview. *Hepatology International*. 2017; 11(6): 485-499. Available from: https://doi.org/10.1007/s12072-017-9830-1.

[20] Richardson N, Wootton GE, Bozward AG, Oo YH. Challenges and opportunities in achieving effective regulatory T cell therapy in autoimmune liver disease. *Seminars in Immunopathology*. 2022; 44(4): 461-474. Available from: https://doi.org/10.1007/s00281-022-00940-w.

[21] Corpechot C, Heurgue A, Tanne F, Potier P, Hanslik B, Decraecker M, et al. Non-invasive diagnosis and follow-up of primary biliary cholangitis. *Clinical Research in Hepatology and Gastroenterology*. 2022; 46(1): 101770. Available from: https://doi.org/10.1016/j.clinre.2021.101770.

[22] Adewoyin O, Wesson J, Vogts D. The PBC model: supporting positive behaviours in smart environments. *Sensors*. 2022; 22(24): 9626. Available from: https://doi.org/10.3390/s22249626.

[23] Daliri A, Asghari A, Azgomi H, Alimoradi M. The water optimization algorithm: a novel metaheuristic for solving optimization problems. *Applied Intelligence*. 2022; 52(15): 17990-18029. Available from: https://doi.org/10.1007/s10489-022-03397-4.

[24] Yalcin AS, Kilic HS, Delen D. The use of multi-criteria decision-making methods in business analytics: a comprehensive literature review. *Technological Forecasting and Social Change*. 2022; 174: 121193. Available from: https://doi.org/10.1016/j.techfore.2021.121193.

[25] Mayo Clinic. *Primary Biliary Cholangitis-Symptoms and Causes*. Available from: https://www.mayoclinic.org/diseases-conditions/primary-biliary-cholangitis/symptoms-causes/syc-20376874 [Accessed 4th October 2024].

[26] Google Search. *Mayo Clinic Study in Primary Biliary Cirrhosis (PBC)*. Available from: https://www.mayo.edu/research/documents/pbchtml/doc-10027635 [Accessed 4th October 2024].

[27] Mohammad-Azari S, Bozorg-Haddad O, Chu X. Shark smell optimization (SSO) algorithm. In: Bozorg-Haddad O. (ed.) *Advanced Optimization by Nature-Inspired Algorithms*. Singapore: Springer; 2018. p.93-103. Available from: https://doi.org/10.1007/978-981-10-5221-7_10.

[28] Freund Y, Mason L. The alternating decision tree learning algorithm. In: *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*. 1999. p.124-133. Available from: https://staff.icar.cnr.it/manco/Teaching/2006/datamining/articoli/Freund_Atrees.pdf [Accessed 26th October 2023].

[29] Biau G, Scornet E. A random forest guided tour. *TEST*. 2016; 25(2): 197-227. Available from: https://doi.org/10.1007/s11749-016-0481-7.

[30] Mohammadifar A, Samadbin H, Daliri A. Accurate autism spectrum disorder prediction using support vector classifier based on federated learning (SVCFL). *arXiv:2311.04606*. 2023. Available from: https://doi.org/10.48550/arXiv.2311.04606.

[31] Alimoradi M, Sadeghi R, Daliri A, Zabihimayvan M. Statistic deviation mode balancer (SDMB): A novel sampling algorithm for imbalanced data. *Neurocomputing*. 2024; 624: 129484. Available from: https://doi.org/10.1016/j.neucom.2025.129484.

[32] Daliri A, Khoshbakhti M, Samadi MK, Rahiminia M, Zabihimayvan M, Sadeghi R. Equilateral active learning (EAL): A novel framework for predicting autism spectrum disorder based on active fuzzy federated learning. In: *Artificial Intelligence and Social Computing*. Huntington, NY, USA: AHFE Open Access; 2024. Available from: https://doi.org/10.54941/ahfe1004655.

[33] Wang X, Xu Y. An improved index for clustering validation based on silhouette index and Calinski-Harabasz index. *IOP Conference Series: Materials Science and Engineering*. 2019; 569(5): 052024. Available from: https://doi.org/10.1088/1757-899X/569/5/052024.

[34] Shahapure KR, Nicholas C. Cluster quality analysis using silhouette score. In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. Sydney, NSW, Australia: IEEE; 2020. p.747-748. Available from: https://doi.org/10.1109/DSAA49011.2020.00096.

[35] Xiao J, Lu J, Li X. Davies bouldin index based hierarchical initialization K-means. *Intelligent Data Analysis*. 2017; 21(6): 1327-1338. Available from: https://doi.org/10.3233/IDA-163.

[36] Akhtar F, Li J, Pei Y, Xu Y, Rajput A, Wang Q. Optimal features subset selection for large for gestational age classification using grid search based recursive feature elimination with cross-validation scheme. *Frontier Computing*. 2020; 551: 63-71. Available from: https://doi.org/10.1007/978-981-15-3250-4_8.

[37] Bej S, Davtyan N, Wolfien M, Nassar M, Wolkenhauer O. LoRAS: an oversampling approach for imbalanced datasets. *Machine Learning*. 2021; 110(2): 279-301. Available from: https://doi.org/10.1007/s10994-020-05913-4.

[38] Pears R, Finlay J, Connor AM. Synthetic minority over-sampling technique (SMOTE) for predicting software build outcomes. *arXiv:1407.2330*. 2014. Available from: https://doi.org/10.48550/arXiv.1407.2330.

[39] de Morais RF, Vasconcelos GC. Boosting the performance of over-sampling algorithms through under-sampling the minority class. *Neurocomputing*. 2019; 343: 3-18. Available from: https://doi.org/10.1016/j.neucom.2018.04.088.

[40] Cao Q, Wang S. Applying over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning. In: *2011 International Conference on Information Management, Innovation Management and Industrial Engineering*. Shenzhen, China: IEEE; 2011. p.543-548. Available from: https://doi.org/10.1109/ICIII.2011.276.

[41] Zeng M, Zou B, Wei F, Liu X, Wang L. Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. In: *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*. Chongqing, China: IEEE; 2016. p.225-228. Available from: https://doi.org/10.1109/ICOACS.2016.7563084.

[42] Douzas G, Bacao F, Last F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*. 2018; 465: 1-20. Available from: https://doi.org/10.1016/j.ins.2018.06.056.

[43] Koziarski M, Woźniak M. CCR: A combined cleaning and resampling algorithm for imbalanced data classification. *International Journal of Applied Mathematics and Computer Science*. 2017; 27(4): 727-736. Available from: https://doi.org/10.1515/amcs-2017-0050.

[44] Chen X, Jeong JC. Enhanced recursive feature elimination. In: *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. Cincinnati, OH, USA: IEEE; 2007. p.429-435. Available from: https://doi.org/10.1109/ICMLA.2007.35.

[45] Rückstieß T, Osendorfer C, Van Der Smagt P. Sequential feature selection for classification. *Advances in Artificial Intelligence*. 2011; 7106: 132-141. Available from: https://doi.org/10.1007/978-3-642-25832-9.

[46] Böhning D. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*. 1992; 44(1): 197-200. Available from: https://doi.org/10.1007/BF00048682.

[47] Chen S, Webb GI, Liu L, Ma X. A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*. 2020; 192: 105361. Available from: https://doi.org/10.1016/j.knosys.2019.105361.

[48] Chang C-C, Lin C-J. Training v-support vector classifiers: theory and algorithms. *Neural Computation*. 2001; 13(9): 2119-2147. Available from: https://doi.org/10.1162/089976601750399335.

[49] Alsmadi MK, Omar KB, Noah SA, Almarashdah I. Performance comparison of multi-layer perceptron (Back Propagation, Delta Rule and Perceptron) algorithms in neural networks. In: *2009 IEEE International Advance Computing Conference*. Patiala, India: IEEE; p.296-299. Available from: https://doi.org/10.1109/IADCC.2009.4809024.

[50] Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*. 1989; 1(2): 270-280. Available from: https://doi.org/10.1162/neco.1989.1.2.270.

[51] Lavin A, Gray S. Fast algorithms for convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE; 2016. p.4013-4021. Available from: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Lavin_Fast_Algorithms_for_CVPR_2016_paper.html [Accessed 26th October 2023].

[52] Deng Y, Ren Z, Kong Y, Bao F, Dai Q. A hierarchical fused fuzzy deep neural network for data classification. *IEEE Transactions on Fuzzy Systems*. 2016; 25(4): 1006-1012. Available from: https://doi.org/10.1109/TFUZZ.2016.2574915.