

Review

Machine Learning (ML) in Medicine: Review, Applications, and Challenges

Amir Masoud Rahmani ^{1,†}, Efat Yousefpoor ², Mohammad Sadegh Yousefpoor ², Zahid Mehmood ³ , Amir Haider ^{4,†} , Mehdi Hosseinzadeh ^{5,*}  and Rizwan Ali Naqvi ^{4,*} 

¹ Future Technology Research Center, National Yunlin University of Science and Technology, Douliou 64002, Taiwan; rahmania@yuntech.edu.tw

² Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful 73210, Iran; eyousefpoor@iaud.ac.ir (E.Y.); ms.yousefpoor@iaud.ac.ir (M.S.Y.)

³ Department of Computer Engineering, University of Engineering and Technology, Taxila 47050, Pakistan; zahid.mehmood@uettaxila.edu.pk

⁴ School of Intelligent Mechatronics Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Korea; amirhaider@sejong.ac.kr

⁵ Pattern Recognition and Machine Learning Lab, Gachon University, 1342 Seongnamdaero, Sujeonggu, Seongnam 13120, Korea

* Correspondence: mehdi@gachon.ac.kr (M.H.); rizwanali@sejong.ac.kr (R.A.N.)

† Amir Masoud Rahmani and Amir Haider have contributed equally to this work.

Abstract: Today, artificial intelligence (AI) and machine learning (ML) have dramatically advanced in various industries, especially medicine. AI describes computational programs that mimic and simulate human intelligence, for example, a person's behavior in solving problems or his ability for learning. Furthermore, ML is a subset of artificial intelligence. It extracts patterns from raw data automatically. The purpose of this paper is to help researchers gain a proper understanding of machine learning and its applications in healthcare. In this paper, we first present a classification of machine learning-based schemes in healthcare. According to our proposed taxonomy, machine learning-based schemes in healthcare are categorized based on data pre-processing methods (data cleaning methods, data reduction methods), learning methods (unsupervised learning, supervised learning, semi-supervised learning, and reinforcement learning), evaluation methods (simulation-based evaluation and practical implementation-based evaluation in real environment) and applications (diagnosis, treatment). According to our proposed classification, we review some studies presented in machine learning applications for healthcare. We believe that this review paper helps researchers to familiarize themselves with the newest research on ML applications in medicine, recognize their challenges and limitations in this area, and identify future research directions.

Keywords: artificial intelligence (AI); machine learning (ML); diagnosis; treatment; medicine



Citation: Rahmani, A.M.; Yousefpoor, E.; Yousefpoor, M.S.; Mehmood, Z.; Haider, A.; Hosseinzadeh, M.; Ali Naqvi, R. Machine Learning (ML) in Medicine: Review, Applications, and Challenges. *Mathematics* **2021**, *9*, 2970. <https://doi.org/10.3390/math9222970>

Academic Editors: Bo-Hao Chen and Amir Mosavi

Received: 3 October 2021

Accepted: 17 November 2021

Published: 21 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial intelligence (AI) means developing computer-based algorithms, which can execute tasks similar to human intelligence. In some medical research, both the terms “artificial intelligence” and “machine learning” may be used interchangeably [1,2]. It is not correct and should be differentiated between the two terms. In fact, artificial intelligence includes a learning spectrum and is not limited to machine learning [3,4]. AI includes representation learning, deep learning, and natural language processing (NLP). AI indicates computational programs, which imitate and simulate human intelligence in problem-solving and the learning process [5,6]. In healthcare, artificial intelligence uses computer algorithms for discovering information from raw data to accurately and correctly make decisions in medicine [7,8].

Machine learning (ML) is a subset of artificial intelligence. It can automatically discover data patterns. ML-based models learn automatically and experimentally and do not need

to be explicitly programmed [9,10]. In other words, the learning model learns based on samples, whereas explicit programming follows rules or a limited hypothesis [11,12]. ML improves efficiency and reliability and reduces costs in computational processes. Moreover, it can accurately and rapidly generate models through data analysis. Machine learning presents tools that can process a large amount of data, the volume of which is far beyond human understanding. For example, health data may include demographic data, images, laboratory results, genomic data, medical records, and data obtained from sensors. Various platforms are used to generate or collect these data samples; for example network servers, electronic health record (EHR), genomic data, personal computers, smartphones, mobile applications, sensors [13,14] and wearable devices [15,16]. Figure 1 represents various data generation resources in healthcare.

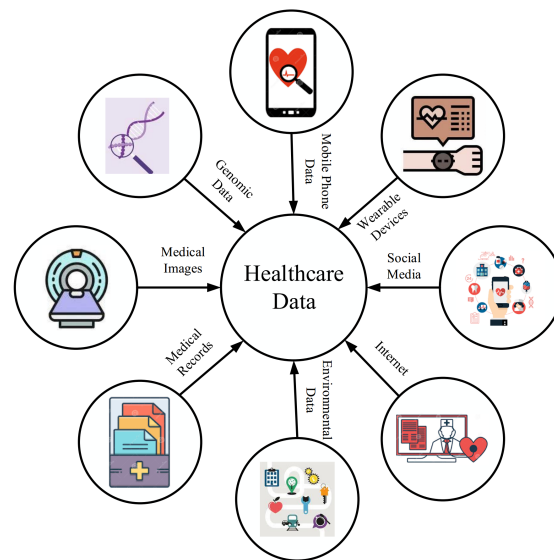


Figure 1. Different data generation sources in healthcare.

Medicine is known as the most important application of artificial intelligence and machine learning [17]. In the mid 20th century, researchers presented many medical decision-making systems. Rule-based methods were very popular in 1970 [18,19]. They were successfully used to interpret electrocardiograms (ECGs), identify diseases and select appropriate treatment methods. However, rule-based systems were costly and highly vulnerable. They need to accurately interpret decision-making rules. They should also be updated continuously. They are known as the first generation of AI-based systems [20,21]. In these systems, medical knowledge must be interpreted accurately by experts to formulate decision-making rules. In contrast, new AI-based models use machine learning (ML) techniques to extract data patterns from complex environments [22,23]. ML has many applications in medicine. These applications include disease identification and classification, the risk ranking of diseases, and the selection of appropriate treatment approaches. Figure 2 displays some ML applications in healthcare. In recent years, researchers have presented many studies that focus on different aspects of healthcare [24,25]. They have used various machine learning methods such as Naïve Bayes (NB), artificial neural networks (ANNs), evolutionary algorithms (EAs), support vector machines (SVMs) and fuzzy systems (FSs) [26], as well as some hybrid methods, such as neuro-genetic systems or neuro-fuzzy systems in their research.

Many researchers work on artificial intelligence and machine learning in healthcare every day. Therefore, we must review more research in this area due to the large advancements in machine learning techniques and their applications in medicine. In Table 1, we present some review papers on ML applications in healthcare. These papers have often focused on ML applications in a specific medical field, for example, medical imaging or machine learning applications for diagnosing or treating a specific illness. They pay less

attention to the structure of ML-based models used in different methods. AI specialists should be aware of the structure of learning models used in different approaches and identify their strengths and weaknesses to improve these models in healthcare. Because there are few review papers, for example [27], in healthcare, which consider the structure of ML-based models. Therefore, this subject requires more attention. Consequently, in this paper, we review the concepts associated with the structure of ML-based models in healthcare and consider their applications in the healthcare field. This paper provides a comprehensive view for artificial intelligence researchers to answer the question, “how can machine learning techniques be used to improve different healthcare methods?” Table 2 compares our review paper with other review papers in this area. In this paper, we first present a classification of machine learning-based schemes in healthcare. This classification categorizes machine learning-based schemes in healthcare based on data pre-processing methods (data cleaning methods, data reduction methods), learning methods (unsupervised learning, supervised learning, semi-supervised learning, and reinforcement learning), evaluation methods (simulation-based evaluation and practical implementation-based evaluation in real environment) and applications (diagnosis, treatment).

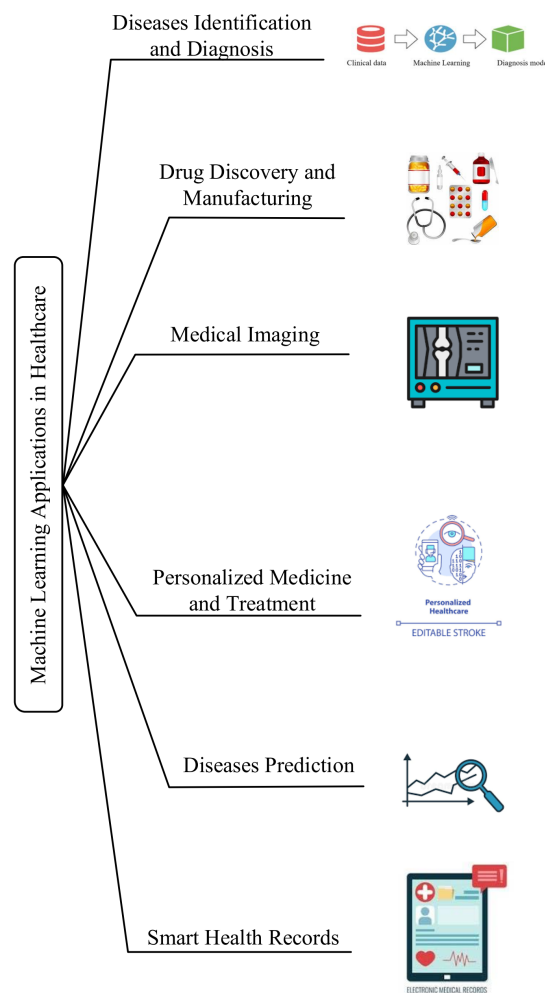


Figure 2. Various ML applications in healthcare.

Table 1. Some review papers on ML applications in healthcare.

Papers	Description
[27]	Miotto et al. [27] first introduced the deep learning framework in summary and expressed its superiority compared to traditional learning methods. Then, they examined some research studies related to the use of deep learning in the healthcare field, specifically its applications in medical imaging, electronic health records, genomics, and mobile apps. They also present challenges and opportunities for deep learning in healthcare.
[28]	Alafif et al. [28] reviewed ML applications on COVID-19 diagnosis and treatment. In this paper, they presented new ML-based methods to diagnose and treat COVID-19. Moreover, they introduced tools and available datasets in this area. In [28], the authors presented some challenges and future research directions in this area. The authors of [28] stated that machine learning can be used for diagnosis, treatment recommendations for controlling disease, drug production, and vaccines. They categorized ML-based methods into two classes: (1) Diagnostic methods that include medical image analysis, non-invasive measurements, and sound analysis; (2) Treatment-based methods include drug development and vaccine development. For more details, please refer to [28].
[29]	Tayarani [29] provided various applications of artificial intelligence-based methods for diagnosis, treatment, monitoring patients, detecting disease severity, digital image processing, drug production, and checking out the outbreak of COVID-19 disease. The proposed classification in this paper includes five Sections (1) Clinical applications of machine learning-based techniques for diagnosis, treatment, and monitoring patients COVID-19; (2) ML applications for chest image processing; (3) Machine learning-based methods for studying Coronavirus and its specifications; (4) Machine learning-based schemes for modeling the COVID-19 outbreak, including epidemic prediction, monitoring pandemic, controlling and managing pandemic; (5) Investigating the dataset available in this area. In this paper, the authors attempted to cover all research works provided in this area. This review paper helps researchers to better manage this disease.
[30]	Smiti [30] examined the main concepts of machine learning in healthcare. In this paper, in the first step, the healthcare process and its various phases are described in summary. According to [30], the healthcare process has four parts: prevention, detection, diagnosis, and treatment. Then, the machine learning process is briefly explained and various machine learning algorithms, including supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning, are presented. Then, the author investigated the ML applications for identifying diseases, producing drugs, doing robot-assisted surgery, and analyzing medical data. In this article, the author specifically focuses on medical data analysis and its challenges in this area.
[31]	Shouval et al. [31] provided various tools for physicians and researchers to achieve a better understanding of machine learning and its applications to hematology. In this regard, they presented some guidelines for designing machine learning-based methods and studied a number of machine learning applications in hematology. Then, the authors introduced types of learning methods, including supervised learning, unsupervised learning and reinforcement learning. In this paper, the authors presented a standard framework for designing machine learning-based models. This framework includes six steps: problem understanding, data understanding, data preparation, data modeling, evaluation, and deployment. Finally, they expressed challenges to and restrictions on machine learning in the medical field and specifically hematology.
[32]	Olsen et al. [32] examined the machine learning algorithms and their applications to heart failure. For this purpose, the authors briefly introduced machine learning and its applications in healthcare. They also presented some important points when designing machine learning-based models. In this paper, machine learning-based methods are divided into three categories based on the learning model: supervised learning, unsupervised learning, and deep learning. Then, machine learning methods are divided into three main categories based on application: diagnosis, classification, and heart failure prediction. Finally, the authors presented challenges and obstacles of machine learning in medicine.

Table 2. Comparison between our review paper and other review papers.

Review Paper	Pre-Processing Methods	Learning Model	Evaluation Criteria	Application
[27]	×	✓	×	✓
[28]	×	×	×	✓
[29]	×	×	×	✓
[30]	×	✓	×	✓
[31]	×	✓	×	×
[32]	×	✓	×	✓
Our survey	✓	✓	✓	✓

We believe that this review paper helps AI researchers to familiarize themselves with the latest research on ML-based approaches in healthcare, recognize the challenges and limitations in this area, and become aware of future research directions. In this review paper, we focus on a number of papers related to machine learning in healthcare published in 2017–2021. We also reviewed and studied various review papers, book captures, research papers, conference papers from different publications such as Springer, Elsevier, IEEE, Wiley, Taylor & Francis, Nature, ACM, and MDPI. Because the number of papers published in the healthcare field is very high, we do not study all of them in the limited volume of this review paper. As a result, we have selected the papers that have recently been published in the healthcare field, provide a more detailed evaluation, and use a larger dataset among papers with the same concept. Then, we remove other papers. We use Google Scholar to find these papers and search various phrases such as “Machine learning”, “Artificial intelligence in medicine”, “Machine learning applications in medicine”, “Intelligent medicine”, “Supervised learning in healthcare”, “Unsupervised learning in healthcare”, “Semi-supervised learning in healthcare”, “Reinforcement learning in healthcare”, “Deep learning”, and “Future hospitals”.

In the following, the paper is organized as follows: in Section 2, machine learning and its applications in healthcare are expressed. In Section 3, we present the general framework for designing a learning model in the medical field. In Section 4, our proposed classification is introduced. In Section 5, we study some ML-based methods in healthcare in accordance with the classification provided in this paper. In Section 6, we summarize discussions about the ML-based methods examined in this paper. In Section 7, we describe some challenges and restrictions on the use of machine learning in medicine briefly. Finally, the conclusion of the paper is presented in Section 8.

2. Machine Learning

Empowering machines for learning like humans is similar to a dream because machines are not inherently intelligent [16,18]. There are some differences between humans and machines when performing their works, one of these differences is intelligence. This means that humans can learn from their previous experiences, but machines do not have this ability. In fact, they must be programmed to follow certain instructions [25,33]. Today, machine learning allows computers to learn from experiences. In the past, traditional computational algorithms included a set of programmed instructions explicitly, which is called “Hard coded”. Computers used these instructions to solve a problem, while today, machine learning helps computers to learn decision-making rules, so that there is no need for programmers who manually develop these rules [34,35]. This is called “Soft coded”. Machine learning is a subset of artificial intelligence (AI). ML-based machines are more intelligent and do not need human intervention. In fact, the term “smart machine” is a symbol [36]. It refers to machine learning and its goals. In 1995, Allan Turing expresses the question for the first time: “Can a machine think?”. He introduced a test called the “Turing Test”. This test evaluates a machine based on intelligence [37,38]. Today, there are various

definitions of machine learning. For example, Arthur Samuel defines machine learning as “a study field that allows computers to learn without explicit programming” [39]. Ethem Alpaydin also defines machine learning as “an area for programming computers based on data samples or experience to improve a performance criterion” [40]. In the phrase “machine learning”, “learning” represents the search process in the possible representation space to create the best representation based on available data [41,42]. Furthermore, “machine” refers to an algorithm that performs search operations. This algorithm is a combination of mathematics and logic [41,42]. In general, the purpose of machine learning is to answer the question: “How can a computer program be made using historical data to solve a problem and automatically improve the performance of the program using experience?” [43,44]. In fact, machine learning is a technology for designing computational algorithms that imitate human intelligence and learn from the surrounding environment. In machine learning, a system is made and trained using a large amount of data (millions of data samples) to manage very complicated tasks. The purpose of this model is to decide, predict or perform tasks without explicit programming. When this model takes inputs, it must be able to produce the desired output. Sometimes, humans can easily understand this model. However, in some cases, it is similar to a black box. This means that humans cannot easily understand this model. In fact, this model approximates the process, which must be imitated by a machine [20,45].

ML Applications in Healthcare

Machine learning has many applications in healthcare. It can facilitate time-consuming and complex tasks in this area. Today, the rapid and significant progress in machine learning (ML), designing faster processors, and accessing digital health data have created opportunities to improve the healthcare process. These new technologies reduce costs, accelerate proper drug discovery, and improve the therapeutic results. Today, machine learning is attracting investors and the main players in the healthcare field [46]. In general, ML applications in the medical area can be divided into three categories:

- **First Category-Improving Available Medical Structures:** These applications are the simplest ML applications in the medical domain. They improve the performance of existing structures [47,48]. These ML-based technologies define specific and rule-based tasks for common applications such as simulation and data confirmation. Classifying digital medical images in healthcare services is one of these machine learning applications. It improves the accuracy of traditional image processing techniques. Machine learning can also be used to analyze radiological images to predict whether there is a particular disease or not. Moreover, ML can be used to evaluate retinal images to determine whether patients are subject to visual threats or not. For example, Aindra is a medical company based on artificial intelligence and machine learning. It uses an ML-based platform to classify medical images. Its purpose is to diagnose cancers in a more accurate and faster manner.
- **Second Category-Upgrading Medical Structures:** In this category, machine learning applications provide structures with new abilities. They move towards personalization. Precision medicine is one of these ML applications [8,49]. It is a kind of medical treatment that targets the specific needs of a person based on her or his characteristics (for example, the genetic arrangement of the person). For example, iCarbonx is moving towards personalized healthcare services. For this, it uses large datasets, biotechnology, and artificial intelligence.
- **Third Category-Independent Medical Structures:** This category of ML applications is expanding recently. They create the ML-based models to perform their actions independently based on pre-defined goals [11]. For example, one of the future applications in the healthcare field is to build a hospital without physicians [37,38]. As a result, we must prepare ourselves for a robotic future based on machine learning and artificial intelligence. Therefore, we must plan the role of robots in future hospitals. In the near future, robots will carry out all healthcare processes from diagnosis to surgery. Today, in developed countries such as China, Korea, and the United States, robots help

surgeons to do surgery in the operating room [50,51]. However, this new technology has some weaknesses and imperfections, but it is rapidly advancing and should still be developed. For example, the Mayo Clinic is moving towards a hospital without doctors. Currently, they design its components. However, these components should be sufficiently tested in terms of various standards. Today, surgeons use robots to improve the surgical process [52,53].

3. The General Framework for Designing a Learning Model in Medicine

In this section, we introduce various phases for designing a learning model in the healthcare field. Note that the purpose of this section is that researchers understand how to design a learning model in medicine. We recommend researchers review and undertake more research in this area to achieve a deep understanding of and knowledge about learning models [18,21]. For designing a learning model in the healthcare field, we must consider five main phases: problem definition, dataset, data preprocessing, ML model development, and evaluation. These phases are shown in Figure 3. In the following, each of these phases is described in detail.

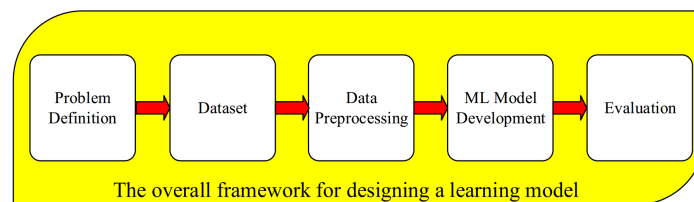


Figure 3. Different phases for designing a learning model.

Problem Definition. When designing a learning model in the healthcare field, we must first answer the question: “*What is the purpose of designing this learning model?*” To design a useful model, the first step is to identify problems and challenges in the healthcare field. Researchers should also analyze exactly how to improve medical services using machine learning. In addition, they should examine the existing solutions presented in this area so far [31]. In the first phase, a key point is to review data availability. This means that researchers should be aware of existing data sources because data should be sufficiently available for developing the learning model and evaluating this model. In the healthcare field, the lack of data can be due to a lack of digital data, patient privacy, commercial issues, or rare diseases.

Database. When designing a learning model in the healthcare field, datasets are used for training, validating, and testing. Healthcare datasets may include demographic information, images, laboratory results, genomic data, and data obtained from sensors [54,55]. Various platforms are used to produce or collect these data, for example network servers, e-health records, genome data, personal computers, smartphones, mobile applications, and wearable devices [56,57]. Today, the Internet and cloud-based technology could improve global connections [58,59]. As a result, data availability has become easier. Before developing a learning model in the healthcare field, it is necessary to design the appropriate mechanism for evaluating the learning model because it is not enough for machine learning for the designer to claim that its learning model has a high performance and is very desirable. ML-based models are data-centric. Therefore, they may be faced with a problem called overfitting or underfitting [60,61]. An efficient learning model should make a tradeoff between overfitting and underfitting. This means that it must have an appropriate bias and proper variance. Underfitting occurs when we design a very simple learning model relative to the complexity of the problem and the size of the dataset. This learning model has a weak performance on both training sets and testing sets. This means that it has a lot of bias. On the other hand, overfitting also occurs when the learning model is very complex and includes large parameters relative to the complexity of the problem and the size of the dataset. In this case, this model has a good performance on the training

dataset, whereas it has a weak performance for the testing set. In this case, it has a high variance. In general, a proper learning model should have low bias and low variance. Figure 4 describes the overfitting and underfitting problems.

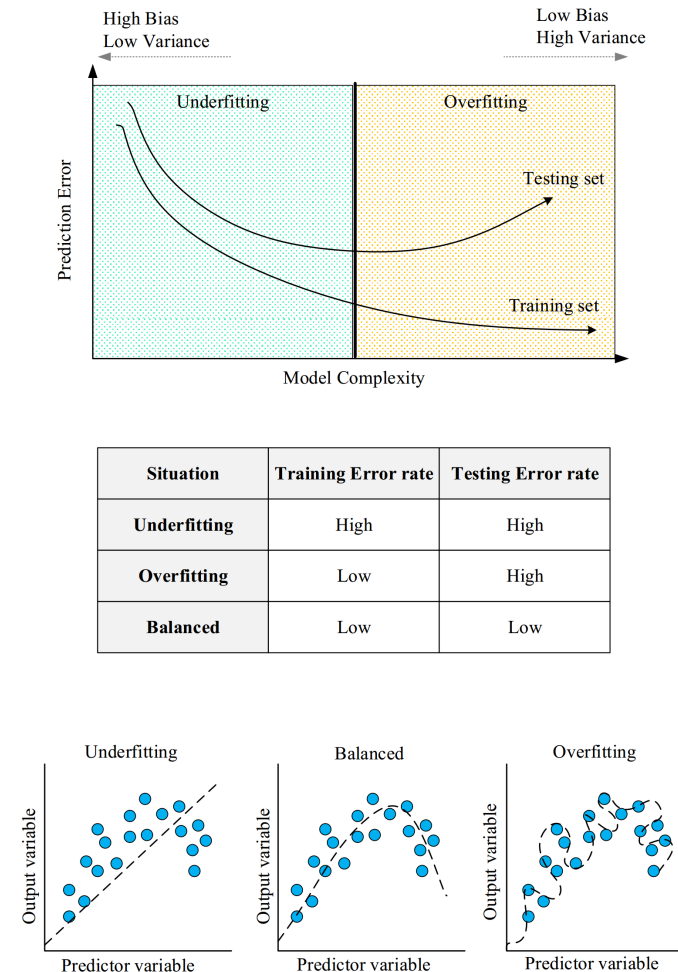


Figure 4. Overfitting or underfitting description.

In order to prevent overfitting, a common solution is that the dataset is divided into two parts: training set and testing set. The “training set” indicates a dataset used for training the learning model and adjusting its parameters. The “testing set” also indicates a dataset used for evaluating the performance of the learning model. Usually, the training set is larger than the testing set, for example, the ratio of 70 to 30. One solution for selecting the training set and the testing set is to randomly divide the dataset into two parts. Another important point is that, sometimes, the dataset is small. Therefore, it is not possible to assign a part of the dataset only for testing. In this case, the K-Fold Cross-Validation technique is used [62,63]. In this technique, the dataset is divided into k sections. Then, a section is used for testing and $k - 1$ sections are used for training. This process is repeated k times so that, in each step, a new section is used for testing. Then, we must evaluate the performance of this learning model in each step. Finally, the overall performance of the learning model is equal to the average performance in k steps. K-Fold Cross-Validation is shown in Figure 5.

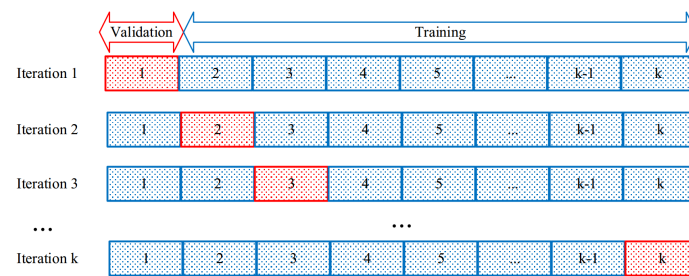


Figure 5. K-Fold Cross-Validation.

Data Pre-Processing. When designing a learning model in the healthcare field, one of the most challenging issues is data preprocessing because a machine learning model requires high-quality data to achieve a higher quality in the training process and a more suitable performance in terms of accuracy. In general, data pre-processing is a process for investigating noisy data, missing values, duplicate data, and contradictory data. The purpose of this process is to increase the quality of the database before creating the learning model. Therefore, in data pre-processing, we may need to filter outliers or estimate missing values. If data also have high dimensions, some data reduction methods, such as feature selection [64,65] or feature extraction [66], can be used. Feature selection selects the best subset of features. On the other hand, feature extraction finds a new dataset with lower dimensions based on the initial data set.

ML Model Development. When designing a learning model in the healthcare field, we must consider the database size, type of learning scheme, and model inference time. We determine the complexity of a learning model based on the database size to avoid overfitting or underfitting. Considering the training time of a learning model is very important. However, learning models with more parameters can produce more accurate results. However, in this case, these models perform more computational operations and need a longer time for training. As a result, they cannot be used for real-time applications. Therefore, lightweight architectures are more appropriate for designing a learning model. Considering the type of learning scheme is also very important when developing ML models [67,68]. In general, there are four main learning methods, including supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [69,70]. We describe these techniques more accurately in Section 4.

Evaluation. Evaluating a machine learning-based system means executing various operations to detect differences between the current behavior of the system and the expected behavior [71]. After designing a learning model in healthcare, the necessary evaluations should be performed to determine an answer to the question, “Does this model have the deployment conditions in real environments?” In the evaluation process, designers use various scales to examine the performance of the learning model. This evaluation determines its strengths and weaknesses. In addition, after deploying the learning model in real environments, we must re-examine the performance of the learning model to evaluate its behavior when interacting with real users [72,73]. Different evaluation aspects of a machine learning system include: evaluating the data used to build the final learning model, evaluating the learning algorithms used to design the final model, and evaluating the performance of the final model. In the following, we explain these aspects more precisely:

- Evaluating the data used to build the final learning model: The performance of learning models depends highly on data. Any error in the data can negatively affect the final model and weaken its performance. In the data evaluation process, it is necessary to answer different questions. For example, are there enough data to train and test the model? Can the existing data be considered representative of all real data for a specific area? Is available data balanced? Is there any hostile or false information in data?
- Evaluating the learning algorithms used to design the final model: At this step, learning algorithms used for creating the final learning model must be carefully evaluated

to determine possible errors in designing or selecting the algorithms. For example, the designer should test different learning algorithms to select the most suitable algorithm for building the final model. When we do not perform sufficient tests to select the proper learning algorithm, it may increase the error rate in the final learning model. In addition, at this step, we can adjust different parameters of a learning algorithm. For example, SVM parameters, or artificial neural networks parameters, such as the number of neurons in each layer, the number of hidden layers and network weights or decision tree parameters, including the number of leaves or its depth.

- Evaluating the performance of the final model: After constructing and training the final model, its performance must be evaluated based on the following factors:
 - Correctness: This factor evaluates how much the current result of the learning system and the expected results are close to each other. In this area, there are evaluation scales listed in the following. For this purpose, we first define some terms:
 - * True positive (*TP*): The number of positive class members, which are properly predicted by the classifier and are labeled as positive class.
 - * True negative (*TN*): The number of negative class members, which are properly predicted by the classifier and are labeled as negative classes.
 - * False positive (*FP*): The number of negative class members, which are falsely predicted by the classifier and are labeled as positive class.
 - * False negative (*FN*): The number of positive class members, which are falsely predicted by the classifier and are labeled as negative class.

In the following, we introduce some important scales for evaluating a learning model. This scales are based on the true positive (*TP*), true negative (*TN*), false positive (*FP*) and false negative (*FN*):

Sensitivity: This scale is defined as a probability so that a classifier truly predicts the result as positive, when the corresponding ground truth is also positive. The other name of this scale is the true positive rate (*TPR*) and it is calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN}. \quad (1)$$

Specificity: This scale is defined as the probability so that a classifier truly predicts the result as negative, when the corresponding ground truth is also negative. The other name of the specificity is the true negative rate (*TNR*) and it is calculated as follows:

$$Specificity = \frac{TN}{TN + FP}. \quad (2)$$

Positive predicted value (PPV): This scale is defined as the probability so that a classifier truly predicts the result as positive, when the test result (output of classifier) is positive. The other name of PPV is precision and it is calculated as follows:

$$PPV = \frac{TP}{TP + FP}. \quad (3)$$

Negative predicted value (NPV): This scale is defined as the probability so that a classifier truly predicts the result as negative, when the test result is negative. This scale is calculated as follows:

$$NPV = \frac{TN}{TN + FN}. \quad (4)$$

Accuracy: This scale is very important. Usually, classifiers are evaluated based on this scale. It is defined as the percentage of samples, which have truly been classified by the classifier. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \quad (5)$$

Matthews correlation coefficient (MCC): It is defined as the correlation coefficient between the predicted result and the corresponding ground truth. It has a value between +1 and -1. If $MCC = +1$, then, this means that the classifier predicts the result truly. If $MCC = 0$, then, this means that the classifier cannot predict the result better than a random manner. If $MCC = -1$, then, this means that there is a full contradiction between the predicted result and the corresponding ground truth. The MCC scale is calculated as follows:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}. \quad (6)$$

False discovery rate (FDR): This scale evaluates the ratio of samples that are falsely predicted as positive, to all samples, which are classified as positive. The FDR scale is calculated as follows:

$$FDR = \frac{FP}{FP + TP}. \quad (7)$$

AU-ROC: This scale is also another important criterion, which is used for evaluating classifiers. It is calculated based on the area under the receiver operating characteristic (ROC) curve. Note that ROC has been drawn based on TPR and FPR. This scale is calculated as follows:

$$AU - ROC = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right). \quad (8)$$

F1-Score: This scale combines two scales, including precision and sensitivity. It is defined as their weighted average. When $F1 - Score = 1$, it is the best value. In contrast, when $F1 - Score = 0$, it is considered as the worst value. This scale is calculated as follows:

$$F1 - Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right). \quad (9)$$

Receiver operating characteristic (ROC) curve: This curve is a method for drawing, organizing and selecting classifiers based on their performance. ROC is a two-dimensional graph. Its vertical axis represents sensitivity and its horizontal axis indicates specificity. A new scale is defined based on ROC called the area under ROC (AUC), which is used for comparing the performance of classifiers. It has a value between 0.5 and one. If AUC is close to 0.5, the classifier has a weak performance.

Note that other evaluation criteria can also be used based on applications [74,75]. For example, ML techniques can be used in applications to automatize tasks such as medical image segmentation. In this case, other scales, such as the Dice coefficient and Jaccard index, can be used to evaluate machine learning models. For more details, refer to [76].

- Model Relevance: This parameter is used to evaluate mismatches between model and data. This refers to overfitting and underfitting. If the available data are not enough, it causes a non-match between the data and the model. The useful solution for solving this issue is cross-validation. However, we do not exactly

know how much overfitting is allowable for the learning model. Suitable methods have been presented in [77,78], for detecting overfitting.

- Efficiency: It represents the prediction speed and the learning speed in a learning model. The efficiency problem occurs when the machine learning-based system conducts the learning or prediction processes very slowly. As a result, ML designers should consider the runtime of learning algorithms.
- Interpretability: Sometimes, learning models are used to decide on medical treatment. As a result, humans must understand the logic and reason behind the decisions taken by these models to trust their decisions so that the final models are socially acceptable. However, it is difficult to define interpretability in terms of mathematics. To understand the interpretability of the ML model, refer to [79]. According to [80], interpretability means the user’s understanding of the decisions taken by ML. Various solutions have also been presented in [81–84] to evaluate the interpretability of a machine learning-based system.

4. Classification of ML-Based Schemes in Healthcare

In this section, we provide a detailed classification of ML-based methods in the healthcare field. This classification, which is also shown in Figure 6, includes four categories:

- Types of data pre-processing methods (data cleaning methods, data reduction methods);
- Types of learning methods (unsupervised learning, supervised learning, semi-supervised learning, and reinforcement learning);
- Types of evaluation methods (simulation-based evaluation and practical implementation-based evaluation in real environment);
- Application (diagnosis, treatment).

In the following, we describe each of these sections exactly.

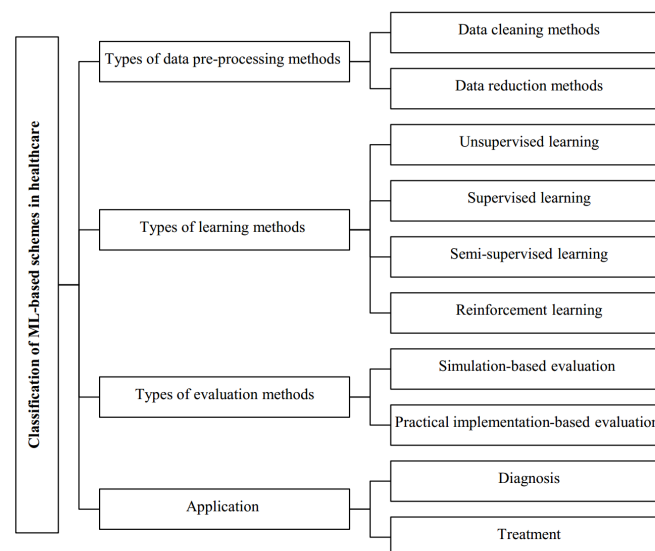


Figure 6. Our proposed classification of ML-based schemes in healthcare.

4.1. Types of Data Pre-Processing Methods

In our proposed classification, ML-based methods in the healthcare field are divided into two main categories based on data pre-processing schemes: data cleaning schemes, data reduction schemes. In the following, each of these methods is explained in detail. Figure 7 also displays types of data pre-processing methods.

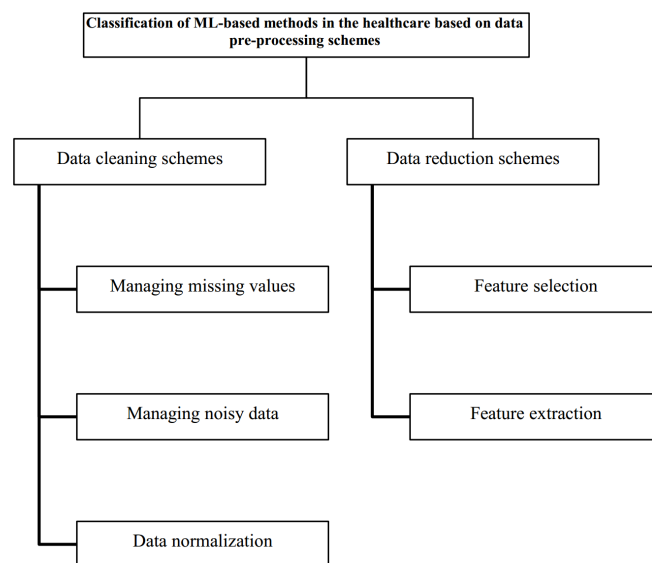


Figure 7. Classification of ML-based schemes in terms of data pre-processing methods.

- Data cleaning methods: Some ML-based methods presented in healthcare use data cleaning methods to eliminate contradictions, such as missing data or noisy data, because such problems are common in the health datasets. These problems have several reasons: (1) Data collection devices are not accurate in the healthcare field. As a result, some data may miss due to the hardware constraints of these devices or some data may be mistakenly recorded; (2) Some data samples are manually produced by physicians or treatment staff. Therefore, they may incorrectly be recorded due to human errors; (3) Some patients inadvertently or deliberately do not express proper information about their illness. This causes errors when recording data. In general, there are several data cleaning methods, including missing value management, noisy data management, and data normalization [18,20].
 - Missing value management: There are two main approaches for managing the missing values in the healthcare field: (1) Removing the data with missing values. Note that if the number of the data with missing values is very high in the dataset, then this approach is not practical; (2) Estimating missing values. Note that if the method used for estimating the missing values is not accurate, then it reduces the accuracy of the learning model.
 - Noisy data management: Filtering methods are used to remove noise in health datasets. This improves the accuracy of the learning model. However, the detection of noisy data is not easy. A solution is to examine the database by professionals and physicians to improve its quality. This causes more accurate modeling and reduces its error. However, this work is costly and time-consuming.
 - Data normalization: Usually, health data are expressed in different scales, for example (age, gender, etc.). We cannot compare these data samples with each other. To solve this problem, a suitable solution is to use the data normalization methods such as the Min–Max method to put data in the range $[0, 1]$.
- Data reduction methods: Often, health data has high dimensions. This weakens the performance of machine learning algorithms because it reduces the quality of the training process and the accuracy of the learning model. Dimensionality reduction means that health data are presented in a compressed form. As a result, this process causes the loss of some information. An appropriate dimensionality reduction scheme in the healthcare field should maintain useful features. Data reduction methods are divided into two main categories: feature selection and feature extraction.

- Feature selection: In this process, a subset of features is selected from the health database to be used in the learning process. The feature selection process is done automatically or semi-automatically [64,65]. Decision-making to remove or maintain a feature is based on the desired application. In general, we categorize feature selection methods into three groups:
 - * Wrapper methods: In these methods, we consider the ML-based model as a black box. Then, we feed this model with different subsets of features. Next, we evaluate its performance for each subset to determine its efficiency. Finally, the best subset of the features is selected. There are two suitable wrapper approaches, including forward selection and backward selection. In the forward selection process, we first consider an empty sub-set. Then, we select a feature of the health database and insert it into the subset. Next, we evaluate the performance of the ML-based model. If it reduces the system error compared to other features, it is added to the final subset. This process continues until the error rate decreases. The backward selection methods are similar to the forward selection approaches. However, there is a difference. In these schemes, we first consider a subset including all features. Then, we select a feature of this subset in each step and remove this feature from the subset. This process continues until the error rate of the learning model decreases [64,65].
 - * Embedded methods: In these methods, the feature selection process is a component of the learning model. For more details, please refer to [64,65];
 - * Filtering methods: These methods are considered as an independent part of the learning model. In these methods, a prioritization test is performed on each feature of the database, so these features are ranked based on a specific criterion. Then, the user chooses the superior features [64,65];
- Feature extraction: These methods are used for compressing health data that have high dimensions [66]. This maintains the main features of the database and removes its noise and correlations. This will accelerate the learning process and produce more accurate results. For example, we introduce some of the most important feature extraction schemes;
 - * Principal components analysis (PCA): PCA is a multivariate and unsupervised technique [18,66]. PCA is tasked to analyze the data for extracting useful information. Then, it displays this information as a set of new orthogonal variables. They are called the principal components;
 - * Linear discriminant analysis (LDA): It is a supervised learning method [18, 66]. Its purpose is to find a linear combination of features, which can be divided into two or more classes. This method tries to maximize the separation between classes and accurately generate linear discriminant functions;
 - * Singular value decomposition (SVD): It is an unsupervised learning technique [18,66]. It is almost close to PCA. In fact, SVD is a generalized version of PCA. It is considered a matrix factorization method and is an efficient scheme for reducing data dimensions. SVD gives an optimal approximation representation of the initial matrix using a low-rank matrix.

4.2. Types of Learning Methods

In our proposed classification, ML-based schemes in healthcare are divided into four main groups based on learning methods: unsupervised learning, supervised learning, semi-supervised learning, and reinforcement learning. Figure 8 shows various learning schemes. Table 3 compares these learning methods.

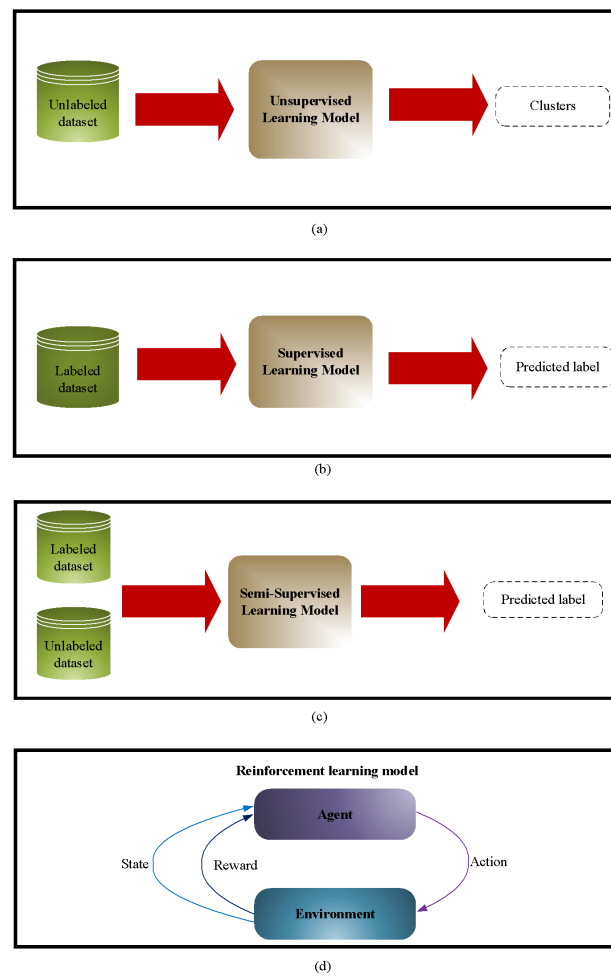


Figure 8. Different learning schemes: (a) Unsupervised learning, (b) Supervised learning, (c) Semi-supervised learning, (d) Reinforcement learning.

Table 3. Comparison of various learning methods.

Scheme	Purpose	Dataset
Unsupervised learning	Identifying data patterns, grouping data samples	Unlabeled dataset
Supervised learning	Predicting the label of the testing set, finding the relationship between inputs and outputs	Labeled dataset
Semi-supervised learning	Predicting the label of the testing set	Both labeled dataset and unlabeled dataset
Reinforcement learning	Finding the best action through interacting with an environment	–

- Supervised learning: In this learning scheme, there is a set of inputs and outputs (labeled databases) [33]. The purpose of this learning technique is to discover the relationship between inputs and outputs in the training process [85,86]. This algorithm produces a function that maps data to labels. Then, it is used to predict the label of unlabeled data. Supervised learning is used when there are outputs (labels) for the training set. In the following, we introduce the most important supervised learning schemes. We express their advantages and disadvantages in Table 4.

Table 4. Supervised learning methods and their advantages and disadvantages.

Algorithm	Advantages	Disadvantages
Naïve Bayes (NB)	Simple implementation, high computational speed, high learning speed, high classification speed, managing overfitting, managing noisy data, and managing missing values	Assuming independence of features, lack of ability to manage features with high correlation, low accuracy
Decision tree (DT)	Simple understanding, high computational speed, high learning speed, high classification speed, managing missing values	The complexity of designing large trees, lack of ability to manage overfitting, low ability to manage noisy data, low ability to manage data with high correlation, medium accuracy
Artificial neural network (ANN)	High flexibility, high accuracy, high classification speed, ability to manage data with high correlation, suitable for nonlinear and complex databases	Difficult implementation, low learning speed, inability to manage missing values, inability to manage noisy data, lack of ability to manage overfitting
Ensemble learning system	High accuracy, ability to manage missing values, ability to manage noisy data, ability to manage overfitting, ability to manage data with high correlation, high classification speed, high stability	Difficult implementation, low learning speed, high computational complexity
Random forest (RF)	Ability to manage noisy data, high classification speed, suitable for large and heterogeneous databases	Difficult understanding by humans, difficult implementation, medium accuracy, low learning speed, low ability to manage missing values, low ability to manage overfitting, low ability to manage data with high correlation
Deep learning (DL)	Suitable for large and high dimensional databases, high accuracy, high classification speed, ability to manage noisy data, ability to manage data with high correlation	Difficult implementation, low learning speed, inability to manage overfitting, low ability to manage missing values
Support vector machine (SVM)	Ability to manage data with linear separability and nonlinear separability, high accuracy, high classification speed, ability to manage data with high correlation	Assuming linear separability for dataset, low ability to manage overfitting, low learning speed, low ability in managing missing values, low ability to manage noisy data
K-nearest neighbor (KNN)	Simple algorithm, stable performance, high learning speed, ability to manage overfitting	High computational overhead, sensitivity to local data structures, medium accuracy, low classification speed, low ability in managing missing values, inability to manage noisy data, inability to manage data with high correlation

- Naïve Bayes (NB): It is a probabilistic classifier, which expresses the relationship between the variables (features) and the target variable (class) as a conditional probability [86,87]. NB is a simple scheme based on Bayes theory. In this method, it is assumed that data points in one class are distributed based on a specific probability distribution. In NB, there is a strong hypothesis called independence of the features. However, this hypothesis is not practical in the real world because most real datasets are strongly correlated. Of course, today, NB has been used to solve many real-world issues and has shown a good performance.
- Decision tree (DT): DT is a supervised learning method. DT constructs the learning model using a set of IF-THEN rules obtained from the training set to predict the output class [88,89]. The hierarchical tree is created based on features in the dataset. In the decision tree, there are three types of nodes: root node (the highest node in the decision tree), the internal node (it indicates an experiment (or comparison) on each feature), leaf node (class label or final result).
- Artificial neural network (ANN): Artificial neural network includes input variables, output variables and weights. The network's behavior depends on the relationship between input and output variables [90,91]. ANNs consist of three layers. Note that each of these layers includes a number of processing units called neurons. The first layer is the input layer that receives raw data. The second layer is also known as the hidden layer that performs a learning task. Note that some ANNs may have several hidden layers. The third layer is also known as the output layer. The output layer depends on the learning process in the hidden layer as well as weights related to input units and hidden units. The designer determines the number of hidden layers and the number of neurons in each layer. This work is conducted through trial and error. Note that there are many approaches for training ANNs and modifying weights to get the lowest error. The most common method is the back-propagation algorithm.
- Ensemble learning system: Ensemble learning systems or multiple classification systems have many applications in a wide range of issues [92,93]. In an ensemble system, different learning methods are combined with each other to improve the prediction result. This helps us to design an exact and robust classification model. The purpose of ensemble learning is to create classifiers with relatively constant bias. Moreover, ensemble learning combines outputs of classifiers through the averaging scheme or other methods to reduce variance and improve accuracy. Ensemble systems are designed in different manners. However, they have three principal parts: (1) Diversity. This means that each ensemble member must be trained using a different method to improve the overall performance of our ensemble learning system. One solution is to select different datasets for training each ensemble member; (2) Training the ensemble members. This part is very important in each ensemble learning system. There are various schemes for training members, for example, bagging and boosting; (3) Combining ensemble members. This refers to a combination rule to obtain a final decision.
- Random forest (RF): This classifier is fast, accurate, and noise-resistant. RF is an ensemble learning technique that classifies data using decision trees [94]. In this scheme, a high number of independent trees are made using an initial training set, for example, an NF matrix, where N is the number of samples, and F indicates the number of features. After creating the random forest, it is used to predict labels. Ultimately, the final label of the samples is calculated using the majority voting.
- Deep learning (DL): It is a supervised learning scheme. It is a subset of ANN. The other name of DL is the deep neural network (DNN) [90]. In DL, there are several hidden layers between the input layer and the output layer. It has several useful benefits. So that, it can extract high-level features from dataset. DL can work with labeled and unlabeled datasets. Moreover, it can be trained to achieve several goals [95].

- Support vector machine (SVM): It is a supervised binary learning scheme. SVM uses the labeled training set to learn the difference between the two classes by mapping input data into a nonlinear feature space [89,96]. In SVM, it is assumed that there is a hyperplane in the feature space. This hypothesis means that data are linearly separable. In the training process, SVM seeks to find this hyperplane, which separates two classes from each other. This hyperplane should have two features: (1) It must exactly separate dataset in two classes; (2) This hyperplane must be in the middle of the two classes to have the highest margin from two classes. However, this hypothesis is not practical. Therefore, SVMs find the best hyperplane that can approximately separate two classes with the least error.
- K -nearest neighbor (KNN): It is the simplest supervised learning method. It is known as a lazy learning scheme [87,88]. In this method, we determine the class of the new sample as follows: first, we compare this sample with the training dataset to determine the k closest samples in the training set. They are called neighbors. In the next step, the class of this data sample is determined based on the majority voting of neighbors. In this method, k is a key parameter that indicates the number of the closest training samples in the feature space.
- Unsupervised learning: In this technique, the dataset includes data samples whose relevant output is not clear [16,33]. This means that data are unlabeled. This learning scheme tries to discover the data patterns and relationships in the data. In unsupervised learning, data are compared based on a similarity scale to be categorized in groups. In the following, we introduce some unsupervised learning methods. We also express their advantages and disadvantages in Table 5.
 - K -means clustering: It is a simple clustering method. The purpose of K -means is to group n data samples to k clusters, so that each cluster is known based on its center. This method is an iteration-based technique [97]. Initially, k random cluster centers is considered and all data points are linked to the closest cluster center. When clusters are established, so that all the data points in the database belong to one of the clusters, a new center will be re-calculated in each cluster. This means that cluster centers are updated in each iteration. This algorithm is repeated until any cluster center does not change.
 - Hierarchical clustering: This clustering scheme aims to group data points to clusters, so that cluster members (data points in a cluster) have the highest similarity to each other compared to data points in other clusters [97]. This process is carried out based on two techniques: top-to-down (Divisive clustering) and bottom-to-up (agglomerative clustering). In the divisive clustering, all data points are first placed in one group. Then, this group is divided into smaller groups. This process continues until each sample is placed in one group. In the agglomerative clustering, each sample is first placed in a cluster. Then, similar groups are merged to establish larger groups. This process continues until all data points are placed in one group. In the hierarchical clustering method, we need no previous information about the number of clusters. This scheme is simply implemented.
 - Fuzzy-c-means (FCM): It is a clustering method based on fuzzy logic. In this method, each sample can be in one or more clusters [97]. FCM determines clusters based on different similarity scales such as distance. Note that one or more similarity scales may be used in the clustering process and this issue depends on application or the dataset. The clustering process is repeated to find best cluster centers. Similar to the K -means clustering method, FCM must be aware of the number of clusters.

Table 5. Unsupervised learning methods and their advantages and disadvantages.

Algorithm	Advantages	Disadvantages
K-means clustering	High clustering speed, suitable for small and large databases, easy understanding	Sensitivity to noisy data, low accuracy, requiring primary knowledge about the number of clusters
Hierarchical clustering	High accuracy, high clustering speed, low sensitivity to noisy data, not needing primary knowledge about the number of clusters, easy implementation	Weak performance for large and small databases
Fuzzy-c-means (FCM)	Low sensitivity to noisy data, high accuracy	Requiring primary knowledge about the number of clusters, low clustering speed, weak performance for small and large databases

- Semi-supervised learning: In this learning method, both labeled and unlabeled datasets are used in the learning process. Therefore, this technique requires a supervised learning algorithm to be trained on a labeled training set. Moreover, an unsupervised learning algorithm should be used to produce data samples with new labels [98,99]. These data samples are added to the labeled training set for the supervised learning algorithm.
- Reinforcement Learning (RL): This learning model allows machines or agents to learn their ideal behavior in a particular situation based on previous experience [12,24]. A reinforcement learning-based model learns continuously through interaction with the environment and collects information to perform its activity [100]. Over time, various methods have been presented to solve the reinforcement learning problem. For example, computational methods such as dynamic programming (DP) to deep reinforcement learning (DRL). In the following, we introduce the most important reinforcement learning methods. We also express their advantages and disadvantages in Table 6.
 - Dynamic programming (DP): It includes a set of methods for calculating an optimal policy of the complete environment model (such as the Markov decision process (MDP)).
 - Monte Carlo (MC) methods: Unlike dynamic programming schemes, the MC-based methods are free-model. This means that they do not require the complete environment model and learn based on experiences (i.e., they learn using interactions with the environment). MC can solve the reinforcement learning problem by averaging sample returns. Monte Carlo (MC) methods guarantee that appropriate sample returns are available because they are often used for episodic tasks. This means that an experience must be divided into episodes. Ultimately, an action is selected and all episodes will also stop. After an episode is terminated, values and policies are updated. Therefore, MC is an incremental episode-by-episode scheme [12,24].
 - Q-Learning: It is known as an appropriate and popular algorithm in reinforcement learning. Q-Learning helps an agent to learn its best actions. In this method, there is a table called Q-Table. This table maintains action-state pairs and the corresponding values. In fact, action-state pairs are known as inputs in this table and the Q-value is its output. In Q-learning, the purpose is to maximize the Q-value [12,24].
 - State-action-reward-state-action (SARSA): It is a reinforcement learning method. Its aim is to learn MDP policy. SARSA and Q-Learning are very similar to each

other. However, there is one main difference between SARSA and Q-Learning. SARSA is an on-policy method. In contrast, Q-Learning is an off-policy method. On-policy means that SARSA follows existing policies to select actions. Then, it updates Q-Value in the Q-Table. Whereas, an Off-policy scheme, like Q-Learning, does not follow the existing policy. It chooses actions using a greedy manner to maximize the Q-Value in the Q-Table [12,24].

- Deep reinforcement learning (DRL): It is a combination of deep learning and reinforcement learning. This scheme can be used to solve many complex issues. It helps the agents to become more intelligent. This improves their ability to optimize the policy. Reinforcement learning is a machine learning technique, which can operate without any database. Therefore, in DRL, agents can first produce the dataset through interaction with the environment. Then, this database is used to train deep networks in DRL [12,24].

Table 6. Reinforcement learning methods and their advantages and disadvantages.

Algorithm	Advantages	Disadvantages
Dynamic programming (DP)	Appropriate convergence speed	Assuming a complete environment model, high computational costs
Monte Carlo (MC) methods	A free-model scheme	Having high variance of returns, low convergence speed, trapping in local optimism
Q-Learning	A free-model, off-policy, and forward learning scheme	Not having a generalization capability, inability to predict the optimal amount for not observed situations.
State-action-reward-state-action (SARSA)	A free-model, on-policy, and forward learning scheme	Not having generalizability, inability to predict the optimal value for unobserved situations.
Deep reinforcement learning (DRL)	Suitable for issues with high dimensions, the ability to approximate the unobserved situations, generalizability	Unstable model, rapid changes in the policy with a slight change in Q-Value

4.3. Types of Evaluation Methods

In our proposed classification, ML-based methods in healthcare are divided into two main categories based on evaluation schemes: simulation-based evaluation and practical implementation-based evaluation. Table 7 compares these evaluation schemes.

- Simulation-based evaluation: Most ML-based models designed in healthcare use simulation tools to evaluate their performance because they are more available than practical implementation. They also have more flexibility and reduce cost. To evaluate ML-based models, it is necessary to simulate this learning model using suitable simulation tools such as MATLAB, WEKA, and R to determine its efficiency. We evaluate these learning models based on various evaluation scales. In general, evaluation criteria are divided into two main categories:
 - Discrimination scales: These scales analyze the ability of an ML-based model for ranking or distinguishing between two classes. The most important discrimination scales are ROC, AU-ROC, F1-Score, Sensitivity, and Specificity. We introduce these scales in Section 3.
 - Calibration scales: These scales determine how many predicted outcomes match actual outcomes. In the real world, these scales are very important because these

scales analyze the expected profits or losses. For example, if the death risk caused by surgery is more than the death risk without surgery, the surgeon may not perform this surgery and abandon it.

- Practical implementation-based evaluation: It is very important to evaluate ML-based models in healthcare using their practical implementation because it allows us to evaluate and analyze learning models in real environments. However, it is very costly because we usually deal with hardware complexities for designing learning models. Repeating scenarios and performing various experiments is also very difficult. In practical implementation, we must evaluate the learning model in a real-time manner and continuously update this model and re-validate it. Some important scales during the practical implementation of learning models in healthcare include their generalizability for new data, user feedback, medical community trust to the designed model, comparing model performance with an expert in the relevant area, and comparing model performance with other existing models.

Table 7. Comparison of various evaluation schemes.

Algorithm	Cost	Availability	Evaluation Result	Implementation
Simulation-Based Evaluation	Low	High	Results May Be Unreal	Easy
Practical implementation-based evaluation	High	Low	Results are more realistic	Complex

4.4. Applications

In our proposed classification, ML-based methods in healthcare are divided into two main categories based on application: diagnosis and treatment.

- Diagnosis: It is a very important stage in the medical field. Machine learning can be used in this area to help physicians and detect the disease in the early stages, and reduce the detection time. For example, machine learning can be used for improving medical images, analyzing laboratory results, segmenting and identifying elements in images, detecting disease, identifying the degree of disease, analyzing signals of devices such as electrocardiography (ECG) for detecting heart failure or electroencephalography (EEG) for evaluating brain activity.
- Treatment: Some ML-based methods can help with the treatment of diseases. For example, machine learning can be used to diagnose suitable doses, personalized therapy, monitoring the treatment procedure, and predicting the progression of the disease. These methods reduce treatment costs, reduce costs related to drug production, improve the treatment procedure, save time to discover appropriate drugs, and solve problems caused by the lack of specialist physicians. Machine learning can also cover the surgical operation to facilitate difficult surgeries with high complexity that are hardly done by humans.

5. Investigating Several ML-Based Methods in Healthcare

In this section, we introduce some ML-based methods in medicine based on the framework provided in this paper and express their weaknesses and strengths. We also review the different sections of each method based on our proposed classification, including data pre-processing scheme, learning technique, evaluation method, and application.

5.1. An Integrated Model Based on LOG and RF

Qin et al. [101] suggested an ML-based method to timely diagnose chronic kidney disease (CKD). First, the authors used the KNN imputation technique to estimate the missing values in the database. They also used optimal subset regression and RF for reducing dimensionality and selecting the most suitable features in the dataset. Then, the learning model was designed using various classifiers. In the following, this learning

model is described in detail. Tables 8 and 9 present the most important characteristics of this ML-based model and its weaknesses and strengths, respectively.

Problem definition. Chronic kidney disease (CKD) is a serious disease, which can threaten general health. ML-based methods can help us to timely and accurately diagnose this disease. In the real world, most medical datasets have many missing values. In [101], the authors believe that existing CKD diagnosis methods have low accuracy, or they used a constrained and weak technique to estimate the missing values. Therefore, the authors of [101] provided an ML-based model for CKD diagnosis. The purpose of this learning method is to increase accuracy and improve its application.

Dataset. In [101], the CKD database available in University of California Irvine (UCI) machine learning repository is used. In this database, there are 400 data points. These data points have 24 features, including 11 numerical features and 13 nominal features. Moreover, there are two final labels, including CKD (In this dataset, there are 250 CKD patients) and NOTCKD (In this dataset, there are 150 data points, which are known as NOTCKD). Note that this dataset is relatively small, and this issue limits the performance of this method in terms of generalizability.

Data pre-processing method. In [101], the KNN Imputation method is applied for estimating the missing values in the database. This method selects k data points without missing values. This data points must be closest to the missing values. Similarity scale is Euclidean distance. Here, there are two cases. One case is that the missing value is a numerical variable. In this case, the missing value is estimated based on the median of k data points. Second case is that the missing value is a nominal variable. In this case, it is obtained based on the majority voting. In addition, this learning model uses a feature selection method based on the optimal subset regression and RF to select the most beneficial features.

ML model development. In [101], a supervised learning scheme is used for predicting CKD disease. In the classification process, various classifiers are examined. The purpose is that classifiers with the best performance are selected for designing the final model. These learning models include: (1) Logistic regression (LOG); (2) Random forest (RF); (3) Support vector machine (SVM); (4) K nearest neighbor (KNN); (5) Naïve Bayes (NB); (6) Feed forward neural network (FNN). Then, they evaluate performance of different models based on several parameters such as accuracy, number of misjudgments, runtime, and among others. Finally, RF and LOG are selected to build the final integration model.

Evaluation. This method uses a simulation-based evaluation. For this, the authors used R 3.5.2 software for simulating the CKD prediction model. To evaluate the learning model, 4-Fold-Cross-Validation method is used. Finally, this learning model has been evaluated according to various criteria such as accuracy, sensitivity, specificity, and F1 Score.

5.2. FCMIM-SVM

Li et al. [102] provided an ML-based system for detecting the heart failure disease. They proposed a feature selection method called FCMIM. In addition, the authors examined different learning techniques, such as artificial neural networks (ANN), support vector machine (SVM), decision tree (DT), Naïve Bayes (NB), K nearest neighbor (KNN), and Logistic regression (LR), for developing the final learning model. Finally, they created the final learning system called FCMIM-SVM. In the following, we describe this ML-based method in detail. Tables 8 and 9 summarize the most important characteristics of this ML-based method and its weaknesses and strengths, respectively.

Problem definition. Heart disease is known to be a serious disease. It can threaten the lives of many people in the world. Traditional methods for detecting this disease are time-consuming, expensive, and inefficient. Therefore, ML-based methods can be very effective because they can detect heart disease using a fast, accurate, and low-cost scheme. In addition, the performance of an ML-based scheme can be improved when a balanced database and an efficient feature selection scheme are used. Regarding the issues

mentioned, the authors of [102] have provided an ML-based method and a feature selection approach to detect heart disease rapidly and accurately.

Dataset. FCMIM-SVM uses a heart disease dataset related to Cleveland. This dataset includes 303 data points. Each data point also has 75 features. There are six data points with missing values. In the pre-processing process, these data points have been removed. Furthermore, there are two classes for the final label: HD or Not-HD.

Data pre-processing method. FCMIM-SVM applies different data pre-processing techniques. For example, it removes data points with missing values from the dataset. It also performs some normalization operations such as Standard Scalar (SS) and Min–Max Scalar on the dataset. Furthermore, FCMIM-SVM designs a feature selection method called FCMIM for reducing dimensionality. Additionally, various feature selection algorithms, such as Relief [103], mRMR [104], LASSO [105] and LLBFS [106], are reviewed.

ML model development. In [102], the authors have first assessed different classifiers like ANN, SVM, DT, NB, KNN, and LR to select the appropriate classifiers for developing the final learning model. Finally, the SVM classifier has been selected by the authors because it has the highest accuracy (i.e., Accuracy = 92.37%). Therefore, the final learning model, called FCMIM-SVM, has been created.

Evaluation. FCMIM-SVM has been evaluated using a simulation-based scheme. This scheme is simulated in Python software. This method also uses the Leave-one-subject-out cross-validation (LOSO) as the evaluation technique. In the evaluation process, the performance of FCMIM is compared with several feature selection approaches. According to the experimental results, the authors believe that FCMIM has a good performance. Then, FCMIM-SVM is evaluated based on various scales such as accuracy, specificity, sensitivity, MCC, and processing time.

Table 8. The most important characteristics of supervised learning-based models.

Scheme	Target	Data Preprocessing Technique	Learning Model	Evaluation Criteria	Simulator
[101]	Diagnosing CKD disease	KNN imputation for estimating missing values and optimal subset regression and RF for selecting useful features	Integrating LOG and RF	Accuracy: 99.83% Sensitivity: 99.84% Specificity: 99.80% F1 Score: 99.86%	R (version 3.5.2)
[102]	Detecting heart disease	FCMIM	SVM	Accuracy: 92.37% Sensitivity: 89% Specificity: 98% MCC: 90%	Python
[107]	Detecting breast cancer	Removing the data points with missing values, finding the importance of each feature using CPG-SVM	Ensemble learning model	Accuracy: 100% Sensitivity: 100% Specificity: 100% Precision: 100% FPR: 0% FNR: 0% F1 score: 100% AUC: 1.0 Gini: 1.0	IBM SPSS Modeler 14.2
[108]	Detecting breast cancer	A feature selection method	Ensemble learning model	Accuracy: 98.07% Precision: 98.10% Recall: 98.10% F1 Score: 98.10% ROC: 97.60%	WEKA 3.9.1
[109]	Detecting, segmenting, and identifying kidney stones	Removing noise, providing a feature extraction method called GLCM, providing a feature selection method	MLP and SVM	Accuracy: 97.5% Prediction rate: 99.7% AUC: 0.922	

Table 9. The most important strengths and weaknesses of supervised learning-based models.

Scheme	Strengths	Weaknesses
[101]	Considering an appropriate method for estimating missing values, performing extensive tests for designing the final model	Considering a small database, not having generalizability, inability to classify severity of the disease
[102]	Considering the data pre-processing process, providing an appropriate feature selection method, comparing the proposed feature selection with other schemes, evaluating different classifiers for developing the final learning model, high accuracy	Considering a small database, not having generalizability, inability to classify severity of the disease, high runtime
[107]	High accuracy, ranking useful features on breast cancer	Considering a small database, not having generalizability, not testing the proposed method with other available databases, not evaluating SVM with different kernel functions, not providing any reason to select SVM and MLP and RBF for designing the ensemble system, not providing a proper data preprocessing method, not reporting the execution time of the final learning system
[108]	High accuracy, performing extensive tests for designing the final model	Not providing a suitable feature selection method, not defining how to select ten features affecting breast cancer, considering a small database, not having generalizability, not testing the proposed method with other available databases
[109]	High accuracy, considering the pre-processing step and providing different pre-processing methods	Not explaining simulation tool, not explaining the dataset, insufficient experiments to evaluate the final model

5.3. CWV-BANN-SVM

Abdar and Makarenkov [107] offered an expert system for detecting breast cancer. This method uses an ensemble learning technique based on support vector machine and artificial neural network. In this method, the optimal parameters of SVM are determined via different experiments. This ensemble system includes two SVMs, multi-layer perceptron (MLP), and radial basis function (RBF) neural network. The performance of neural networks is also improved using boosting technique. In the following, we describe this learning model exactly. In addition, Tables 8 and 9 express the main characteristics of the CWV-BANN-SVM method and its advantages and disadvantages, respectively.

Problem definition. Breast cancer is the most common cancer in the world. This disease requires high costs for treatment. Therefore, ML-based solutions can reduce these costs and increase the accuracy of diagnosis. In general, learning methods reduce the diagnosis time and increase its accuracy. As a result, in [107], an ensemble learning method has been developed to timely and accurately diagnose breast cancer.

Database. In [107], the authors used the Wisconsin breast cancer dataset (WBCD). WBCD has 699 data points. There are two labels for output result, including benign and malignant. Each data point has 10 features. There are 452 data points belonging to the benign class and there are 241 data points belonging to the malignant class.

Data pre-processing method. In the dataset, there are 16 data points with missing values that are removed in the data pre-processing process.

ML model development. To develop the learning model, first, the authors tested a simple SVM with different parameters to find its most appropriate parameters. These parameters include regularization parameter (C), gamma parameter (γ), and ϵ . The authors believe that this improves the accuracy of the learning model and prevents overfitting. For designing the final learning model, the authors performed four main steps. First, they tested six classifiers: simple SVM, polynomial SVM, simple MLP, simple RBF, boosting MLP, boosting RBF. According to the experimental results, the authors selected two polynomial SVMs, boosting

MLP, and boosting RBF to design the final ensemble model. They also applied SVM-CPG to determine the importance of each feature in the database for detecting breast cancer. In the second step, a data pre-processing process is performed for removing data with missing values. In the third step, the selected classifiers are re-evaluated on the modified database. In the final step, the authors created an ensemble classifier using two SVMs, boosting MLP, and boosting RBF. This ensemble system uses the confidence-weighted Voting (CWV) technique.

Evaluation. The CWV-BANN-SVM method uses a simulation-based evaluation. This scheme is simulated in IBM SPSS Modeler 14.2 software. The dataset is divided in two parts so that 50% is used for training and 50% is applied for testing. In the evaluation process, various criteria such as accuracy, sensitivity, specificity, precision, FPR, FNR, F1 Score, AUC, and Gini Index are considered.

5.4. Nested Ensemble Method (NE)

Abdar et al. [108] introduced the nested ensemble (NE) method for automatically predicting breast cancer. NE is a two-layer scheme, which includes classifiers and meta-classifiers. In the following, we explain this method based on our proposed classification in this paper. Table 8 summarizes the most important features of the NE method. Furthermore, Table 9 describes its advantages and disadvantages.

Problem definition. Breast cancer is the most common cancer among women. There are some schemes such as mammography for detecting breast cancer, but they are not accurate. In addition, physicians and specialists such as radiologists, hematologists, and pathologists must cooperate with each other to achieve a precise diagnosis about the disease. This is a very time-consuming work. Therefore, ML-based models can be very beneficial to accurately and rapidly detect this disease. In [108], a ML-based method was presented to automatically diagnose breast cancer. The purposes of this method are to improve accuracy and reduce the required time for detecting malignant tumors.

Database. In [108], NE uses the breast cancer Wisconsin diagnostic database (WDBC). This database includes 256 data samples. Each data sample has 32 features. There are two output labels, including benign and malignant.

Data pre-processing method. In this scheme, a feature selection method has been used to reduce dimensionality. In this process, 10 useful features are selected for detecting breast cancer. Note that the authors do not mention what feature selection method is used in NE, and this process is very ambiguous.

ML model development. To design the NE method, several ensemble learning techniques and some basic algorithms are used. The basic algorithms used in this method include Bayesian network (BN), Naïve Bayes (NB), Stochastic gradient descent (SGD), J48, REP-Tree, and logistic model trees (LMT). In general, NE includes classifiers and meta-classifiers. The meta-classifier includes two or more different classifiers. To develop the final learning model, four nested ensemble learning models are created using stacking and voting techniques (SV). These NEs are:

- SV-BayesNet-2MetaClassifier: BN + LMT + SGD + 2-MetaClassifier (SGD + J48)
- SV-Naïve Bayes-2MetaClassifier: NB + LMT + SGD + 2-MetaClassifier (SGD + J48)
- SV-BayesNet-3MetaClassifier: BN + LMT + SGD + 3-MetaClassifier (SGD + J48 + REPTree)
- SV-Naïve Bayes-3MetaClassifier: NB + LMT + SGD + 3-MetaClassifier (SGD + J48 + REPTree)

Then, these NEs are tested based on different experiments. According to the experimental results, the authors selected *SV-Naïve Bayes-3MetaClassifier* as their final learning model.

Evaluation. In [108], the authors used the simulation-based evaluation. They used WEKA 3.9.1 simulator for implementing NEs. To evaluate these methods, the 3, 5, 10-Fold Cross-Validation technique has been used. NEs are evaluated based on different criteria, including accuracy, precision, recall, F1 Score, ROC, and processing time.

5.5. HMANN

Ma et al. [109] suggested an improved neural network called HMANN. This scheme is used for detecting, segmenting, and identifying chronic renal failure. HMANN is imple-

mented on the Internet of Medical Things (IoMT) platform. This method combines support vector machine (SVM), multi-layer perceptron (MLP), and backpropagation algorithm (BP). In the following, we explain HMANN in detail. Moreover, Table 8 provides the most important characteristics of HMANN and Table 9 expresses its weaknesses and strengths.

Problem definition. When kidneys do not work well, this issue can threaten human life. Therefore, it is very important to timely detect kidney stones. Often, digital images have low contrast. They are also highly noisy. Therefore, it is very difficult to use these images for detecting kidney abnormalities. Artificial neural networks are one of the most common tools for solving this problem. Because they are fault-tolerant. They can also be generalized easily. Moreover, they have a suitable learning ability. Therefore, in [109], a neural network-based system has been developed.

Database. The authors use images in the UCI chronic kidney disease dataset to train and test HMANN. In this method, there is no explanation about this database. The authors do not mention the number of images in the dataset and their type.

Data pre-processing method. As mentioned earlier, digital images often have noise and low contrast. Their evaluation is difficult. In HMANN, the authors have reduced noise using threshold wavelet coefficients. In general, a pre-processing process is performed on these images to overcome the low contrast and noise. The data pre-processing process includes three steps: (1) Rebuilding images using a level set method; (2) Sharpening or smoothing using a Gabor filter; (3) Improving contrast using a histogram equalization process. In addition, a specialist physician performs manually the segmentation process on normal and abnormal digital images. Then, HMANN uses a feature extraction process called the gray-level co-occurrence matrix (GLCM) on these segmented regions to extract features related to this disease. These features include adaptive, Haralick, and histogram features. Then, a feature selection process is performed for selecting nine features.

ML model development. In [109], the final learning model is built based on three main components, including SVM, MLP, and BP. The final learning model is called HMANN. The purpose of HMANN is to classify digital images modified in the previous step, identify kidney stones, and accurately detect their location.

Evaluation. HMANN uses simulation-based evaluation. This method is simulated and evaluated through various experiments to determine its efficiency. However, the authors do not explain the simulation tool, training set, testing set, and other simulation parameters. HMANN is evaluated based on various criteria such as prediction rate, AUC, accuracy, computational time, and ROC.

5.6. SRL-RNN

Wang et al. [110] proposed an ML-based model called SRL-RNN. This scheme uses reinforcement learning and recurrent neural network (RNN). The purpose of SRL-RNN is to solve the dynamic treatment regime (DTR) problem. The main idea of this method is to combine two signals, including indicator and evaluation simultaneously. In the following, we describe SRL-RNN in detail. The most important features of SRL-RNN are represented in Table 10. Furthermore, Table 11 expresses its strengths and weaknesses.

Problem definition. Many researchers reviewed drug recommendation systems to help physicians for better decision-making. These systems can be designed using supervised or reinforcement learning algorithms. Supervised systems utilize similarities between patients to produce recommendations. However, these methods cannot directly learn the relationship between illness and drugs. These methods depend on the ground truth. However, there is no response to this question: *how is this ground truth created?* In this case, they work based on the indicator signal. While reinforcement learning-based systems do not have this problem. However, they may present treatment recommendations that are strongly different from the prescription recommended by the physician. This is because a supervisor does not control them. This problem can increase the treatment risk. In fact, they work based on the evaluation signal. Therefore, the authors of [110] combine supervised

learning and reinforcement learning to produce a new model called SRL-RNN. This method can avoid unauthorized risks and deduce optimal and dynamic treatment.

Database. The authors utilize a large and available database called MIMIC-3 v1.4 to evaluate SRL-RNN. This database includes information about 43,000 patients in the intensive care units (ICU). This information has been collected from 2001 to 2012. It contains information about 6695 specific diseases and 4127 drugs.

Data preprocessing method. In [110], when a data point has many missing values, more than 10 features, then this data point must be removed from the database. On the other hand, when a data point has a small number of missing values, then these missing values are estimated using the KNN method.

ML model development. In [110], the authors presented a deep architecture called SRL-RNN for managing a DTR, including several diseases and different prescriptions. The aim is to learn the prescriptive policy by combining the index signal and the evaluation signal. SRL-RNN includes three main networks: (1) Actor network for producing drugs in a time-variant manner based on the dynamic status of patients. In this process, doctor's decisions play the role of an indicator signal. This means that there is a supervisor to ensure safe actions and speed up the learning process; (2) Critic network for assessing the action related to the actor network to reward or penalize the recommended treatment; (3) LSTM network for developing SRL-RNN to manage a partially-observed Markov decision process (POMDP). It summarizes the observations to produce a more complete observation. Note that LSTM is one of the most famous recurrent neural networks (RNNs). It is known as a deep neural network.

Evaluation. SRL-RNN uses both evaluation methods i.e., simulation-based and practical implementation-based. In the practical implementation, the prescriptions produced by this method are evaluated for two patients in ICU. Note that the authors do not mention the software used to simulate this method. The dataset is divided into three groups, including the training set (80% of the dataset), validation set (10% of the dataset), and testing set (10% of the dataset). In [110], the mortality rate is considered as an evaluation scale to evaluate the effect of this method for reducing mortality. The Jaccard coefficient has been used to measure the compatibility between prescriptions recommended by SRL-RNN and prescriptions produced by the physician.

5.7. A Closed-Loop Healthcare Processing Scheme

Dai et al. [111] simulated the human body using deep neural networks (DNNs) and utilized deep reinforcement learning (DRL) to find suitable treatment schemes for the simulated body. In this method, the simulated body plays the role of a patient and DRL plays the role of a physician. In the following, we describe this scheme exactly. Furthermore, Table 10 expresses the main characteristics of this method and Table 11 presents its advantages and disadvantages.

Problem definition. In healthcare, it is necessary that the human body is continuously monitored to timely perform the corresponding treatments. However, it is not true to perform unauthorized tests on the human body. Therefore, it is necessary to design a virtual human body. However, the human body is a very complex system. Today, modern science has been accompanied by great progress. However, it cannot completely imitate the human body. A solution is to consider the body as a black box to interpret output data in response to input data. This means that it is based on a data-driven method. DNN is a useful tool for modeling the human body because it has a global approximation capability. Therefore, in [111], DNN is used to simulate the human body.

Database. In [111], the authors use a database including 990 tongue images. These images include 9 different structures to train a deep neural network (DNN). Note that the authors do not present exact explanations for the database.

Data pre-processing method. There is no pre-processing method in this scheme.

ML model development. The learning model presented in [111] includes two main components: simulated body and treatment part. The simulated body consists of two main

parts, including regulating network and decoding network. The regulating network is tasked to show the effect of treatment on the health status. Furthermore, the decoding network is tasked to transform a space with low dimensions (i.e., the health status) into a space with high dimensions. In [111], LSTM has been used as a deep learning method for simulating the human body. In [111], the conceptual alignment deep auto-encoder (CADAЕ) has been used as a decoding network. The second component i.e., treatment part is also responsible for receiving observations and producing therapeutic recommendations. This component dynamically interacts with the simulated body. It has two main parts: disease diagnosis and proper therapeutic recommendation. In [111], the author used a deep reinforcement learning (DRL) scheme to merge these two parts. In this regard, they used a deep Q-network (DQN) for discrete space and the deep deterministic policy gradient (DDPG) for continuous space.

Evaluation. This method uses a simulation-based evaluation. Therefore, this scheme is simulated using TensorFlow installed on Python. The simulated body is trained using CADAЕ. This method is evaluated in terms of convergence rate and mis-diagnostic rate. Note that this method has presented the experimental results in a graph form. As a result, we do not present numerical results for this scheme.

5.8. GAN + RAE + DQN

Tseng et al. [112] provided a deep reinforcement learning scheme for making treatment decisions. This method includes three components: (1) GAN for generating artificial data based on a small dataset. (2) Transition DNN for constructing the virtual radiotherapy environment. (3) DQN for determining the optimal radiation dose for the radiotherapy treatment process. In the following, we describe this method in detail. In addition, we present the most important specifications of this method in Table 10. Table 11 describes its strengths and weaknesses.

Problem definition. Usually, doctors believe that surgery is not a suitable option for treating non-small-cell lung cancer (NSCLC) patients and it is better to treat them using radiotherapy. However, this technology is progressing every day. However, its treatment results are not satisfactory. A suitable option is to increase the radiation dose in radiotherapy for enhancing the treatment process. Although, this can increase inflammation due to radiation and reduce the life quality of patients. This research tries to respond to this question: “Whether the machine learning algorithms can determine the optimal radiation dose based on features of patients for controlling tumors locally and minimizing inflammation?” In recent years, deep reinforcement learning has been successfully used in various areas. This is because this learning technique can extract high-level features directly from raw data. Therefore, in [112], DQN is used to determine the radiation dose in radiotherapy.

Database. This research uses a database including 114 NSCLC patients. Note that each data sample data consists of 297 features. For more details, please refer to [112].

Data pre-processing method. In [112], the authors use a feature selection scheme for selecting nine important features to simulate the radiotherapy environment. For this purpose, Bayesian network graph theory is used to hierarchically determine relationships between features and the desired output. This scheme tries to find the minimum features for controlling the tumor locally and reducing inflammation due to radiation.

ML model development. In [112], the authors simulated the radiotherapy environment to design an artificial radiotherapy environment. The transition DNN algorithm is tasked to perform this work. For this work, they used GAN along with the transition DNN algorithm. This is because the available database is very small. As a result, GAN, which is a deep neural network, can produce artificial data very similar to real data. Then, the transition DNN algorithm is trained based on both real data and artificial data to simulate the radiotherapy environment. Next, DQN interacts with this simulated environment to imitate the doctor’s decision and determine the radiation dose for each patient.

Evaluation. This method uses simulation-based evaluation. It applies the MATLAB software for the feature selection process. In this case, AUC is considered as an evaluation

scale. Note that the evaluation process uses a 10-Fold Cross-Validation method. Then, the final learning model is implemented in TensorFlow. As mentioned earlier, there are 114 data samples in the database. Then, GAN uses this database to produce artificial data. After executing this process, 4000 artificial data samples are produced. As a result, the number of data samples (real data and artificial data) is equal to 4114. Then, the DNN algorithm is trained according to this new database. In this case, the evaluation criterion is the average accuracy. Then, the DQN algorithm is executed on 34 patients in the UMCC protocol. In this case, the root mean square error (RMSE) is considered an evaluation scale, which is approximately 0.76.

5.9. HQLA

Khalilpourazari and Hashemi [113] offered a reinforcement learning-based algorithm called HQLA. This algorithm uses the Quebec database to predict the Coronavirus prevalence. In this algorithm, the authors utilize two techniques, including reinforcement learning and evolutionary algorithms. In the following, we describe this method in detail. Table 10 represents the most important features of this method in summary. Furthermore, Table 11 expresses its advantages and disadvantages.

Problem definition. Modeling and predicting the COVID-19 epidemic process can help specialists in the healthcare field to finish its prevalence. However, it is very challenging to predict the COVID-19 prevalence due to its unclear and complex nature. The metaheuristic algorithms are very flexible and efficient. They can solve many problems in healthcare because they reduce computational costs and time complexity. They can also efficiently explore optimal responses. In addition, reinforcement learning algorithms can solve many issues in the real world, especially in healthcare. According to this issue, in [113], the authors combine the metaheuristic algorithms and reinforcement learning to predict the coronavirus pandemic.

Database. Quebec is one of Canada's provinces. The dataset includes data samples related to COVID-19 and the mortality rate recorded from 25 June to 19 July in 2020. This database includes 63713 data samples related to COVID-19 patients and 5770 data samples related to the dead individuals due to COVID-19.

Data pre-processing method. In [113], there is no data pre-processing process.

ML model development. This method (HQLA) combines reinforcement learning and evolutionary algorithms. This scheme can solve complex optimization problems in a short-term time period. HQLA uses various evolutionary algorithms such as GWO [114], SCA [115], MFO [116], PSO [117], WCA [118], and SFS [119] to update the particle position in response space. Q-Learning is used to select the best operator (evolutionary algorithm) in the optimization process to obtain the best efficiency. Q-learning starts with several random operations. Then, it evaluates the efficiency for each operator in each step. This helps Q-Learning to learn the best operations for getting the best response. If an operator improves the final response quality, Q-learning rewards this operator. Otherwise, it penalizes the current operator.

Evaluation. HQLA uses simulation-based evaluation. Note that the authors do not mention the software used to implement this method. In the evaluation process, the mean square error is considered as the objective function. Its optimal amount is equal to 6.26×10^{-6} . The authors also presented several graphs, including convergence rate, a comparison between predicted data and actual data. Evolutionary algorithms have been evaluated in terms of various parameters. It is outside the field of this paper. For more details, please refer to [113].

5.10. tVAE

Baucum et al. [120] introduced the transitional variational auto-encoders (tVAE). It tries to learn the disease progression procedure to map a patient's status to his next state at the next time point. In the following, we present this method in detail. In Table 10, some features of tVAE are expressed. Table 11 presents its advantages and disadvantages.

Table 10. The most important characteristics of reinforcement learning models.

Scheme	Target	Data Preprocessing Technique	Learning Model	Evaluation Criteria	Simulator
[110]	Generating treatment recommendations for DTR	Removing some data points with high missing values and estimating some data points with the small number of missing values	Deep reinforcement learning and recurrent neural network	Jaccard Coefficient: 0.409 Mortality Rate: 0.157	–
[111]	Designing a virtual body and a virtual doctor	–	Deep reinforcement learning and recurrent neural network	–	Python
[112]	Designing a virtual radiotherapy environment and determining the appropriate radiation dose for treating lung cancer	Bayesian network graph theory for selecting useful features	Deep reinforcement learning and recurrent neural network	Accuracy: 100% RMSE: 0.76	TensorFlow
[113]	Predicting the COVID-19 epidemic process	–	Reinforcement learning	MSE: 6.29×10^{-6}	–
[120]	Simulating artificial patients and simulating the virtual treatment policy	Estimating missing values using the sample-and-hold interpolation method and an artificial neural network	Deep reinforcement learning and artificial neural network	MAE: 12.15	TensorFlow

Table 11. The most important strengths and weaknesses of reinforcement learning models.

Scheme	Strengths	Weaknesses
[110]	Using both practical implementation-based evaluation and simulation-based evaluation, using a large database for training and testing this learning model	Not explaining the simulation tool, insufficient experiments to evaluate the final model, not considering a suitable pre-processing scheme
[111]	Using both deep learning and reinforcement learning to design the final learning model, merging two processes, including disease diagnosis and treatment recommendation	Insufficient experiments to evaluate the final model, not considering a suitable pre-processing scheme, not testing the proposed method with other available databases
[112]	Using both deep learning and reinforcement learning to design the final learning model, increasing the dataset size using GAN, using a suitable feature selection method	Insufficient experiments to evaluate the final model, not having sufficient data to evaluate the final model, high error rate
[113]	Using both reinforcement learning and metaheuristic algorithms to design the final learning model, selecting the best operator using Q-learning	Insufficient experiments to evaluate the final model, not considering a suitable pre-processing scheme, not considering different parameters such as sex, age, and among others for designing the final learning model
[120]	Ability to use continuous data samples, considering a random policy when designing the model, using the on-policy reinforcement learning method	Insufficient experiments to evaluate the final model, not considering a suitable pre-processing scheme, not considering the effect of different parameters on the final learning model, not measuring the effect of dataset size on the performance of the learning model, not using different reinforcement learning algorithms to evaluate the performance of the learning model

Problem definition. Reinforcement learning (RL) is a useful tool for developing a personalized treatment regime. For ethical reasons, RL agents cannot directly interact with real patients. Two solutions to this issue are: (1) Training the model using the existing dataset (Off-policy RL); (2) Learning a virtual environmental model using the available dataset (On-policy RL). In [120], the authors presented a deep reinforcement learning method called tVAE. This scheme is based on the on-policy technique. tVAE seeks to learn the disease model accurately.

Database. In [120], the authors used the MIMIC database. It includes information about 2067 patients in ICU. In this database, patients' parameters such as heparin dose and aPTT have been measured every hour. Note that, in this dataset, 42.4% of patients are women. The mean age of the patients is 70.4 and their average weight is equal to 173 Lbs.

Data pre-processing method. The MIMIC database includes missing values. In [120], the sample-and-hold interpolation method is used to determine the missing values related to the heparin dosage. An artificial neural network is used for estimating the missing values corresponding to the aPTT parameter. Note that the authors have normalized all variables in the dataset, but they do not mention the normalization method.

ML model development. tVAE method uses the standard VAE structure for simulating transitions between successive patient states. In this scheme, the purpose is to model a virtual patient environment to learn the prescriptive policy. Next, tVAE trains an artificial neural network so that it receives the continuous latent states as input and produces an output. This method can consider a continuous disease space and create randomness in the model. tVAE is suitable for medical time series. After designing a virtual patient environment, an on-policy reinforcement learning algorithm called A3C is used to learn the best heparin dose.

Evaluation. In [120], tVAE uses simulation-based evaluation. This method is simulated in TensorFlow. In the evaluation process, the dataset is divided into two parts: training set (85% of the data samples) and testing set (15% of the data samples). In addition, the evaluation criterion is the mean absolute error (MAE).

5.11. TE-DLSTM

Zhu et al. [121] presented a semi-supervised learning method called TE-DLSTM to identify body activities using inertial sensors. This method uses a deep long short-term network (DLSTM) to extract high-level features. In the following, we explain TE-DLSTM in detail. Tables 12 and 13 represent the most important characteristics of this method and its advantages and disadvantages, respectively.

Problem definition. Human activity recognition (HAR) is a very important issue for informatics applications, especially healthcare. For example, when users use smartphone applications, HAR helps us to understand their behavior. In fact, HAR discovers their health status and presents high-quality health recommendations. However, a challenging issue is that we deal with unlabeled data when designing the HAR system. One effective solution for this issue is semi-supervised learning. Today, many methods use semi-supervised learning techniques to identify body activity. However, they can only extract low-level and simple features and do not have an acceptable performance. Accordingly, in [121], a DLSTM-based method is presented for designing HAR to extract high-level features.

Database. In [121], the authors used the UCI database, which includes time-series samples collected from 30 people. Their ages are between 19–48 years. Each time-series sample is sampled based on an overlapping window frame, which is equal to 2.56 s. The total number of samples is 10,000. Note that in this database, each data sample has 561 features.

Data pre-processing method. In [121], the authors perform a simple feature extraction process on the database to extract some simple statistical features such as maxim, minimum, mean, and variance. Then, these low-level features feed the neural network to learn high-level features. Note that the final learning model is also a feature extraction method for extracting high-level features from the database.

ML model development. The database used for designing the learning model includes both labeled data and unlabeled data. For developing the learning model, in the first step, an augmentation technique enlarges the database. This technique acts as a regularizer in terms of randomness. Then, the authors extract simple features from the dataset. DLSTM is trained based on these low-level features. Then, the Dropout network acts as a regularizer to enhance the generalization ability of DLSTM. In the next step, the cross-entropy method is used for measuring supervised learning loss. It analyses the difference between the ground truth and the predicted label. The Square Loss method is used for measuring unsupervised learning loss so that the predicted output is compared with the previous ensemble output. Finally, the final loss is calculated based on a combination of supervised learning loss and unsupervised learning loss to obtain deep learning parameters based on the back-propagation method.

Table 12. The most important characteristics of semi-supervised learning-based models.

Scheme	Target	Data Preprocessing Technique	Learning Model	Evaluation Criteria	Simulator
[121]	Extracting high-level features using a semi-supervised learning technique	Feature extraction method	Semi-supervised learning method and deep neural network	Accuracy: 97.21% RunTime: 2.118 s	Python
[122]	Extracting ADR mention from Twitter	Data normalization method	Semi-supervised learning method and deep neural network	F1-Score: 75.1% Precision: 73.1% Recall: 77.4%	Python
[123]	Detecting Normal beats, SVEB, and VEB based on the unlabeled dataset	Data normalization method	Semi-supervised learning method and CNN	SVEB (%) Accuracy: 97.4% Sensitivity: 93.38% Specificity: 97.2% PPR: 59% F1-Score: 72.5% VEB (%) Accuracy: 98.6% Sensitivity: 87.5% Specificity: 99.4% PPR: 90.9% F1-Score: 89.2%	MATLAB
[124]	Segmentation of retinal fundus images	Data normalization method, increasing data samples	Semi-supervised learning method, deep neural network, transfer learning	DRISHTI dataset DSC: 0.967 Accuracy: 0.9957% Jaccard: 0.9314 Sensitivity: 0.9539 Specificity: 0.9993 RIM-ONE dataset DSC: 0.902 Accuracy: 0.9945% Jaccard: 0.8824 Sensitivity: 0.873 Specificity: 0.9981	TensorFlow tool in Python
[125]	Designing a clinical decision support system	Increasing data samples, the feature selection process	Semi-supervised learning method	Balanced dataset Accuracy : 90.9% Unbalanced dataset Accuracy : 87.2%	MATLAB

Table 13. The most important strengths and weaknesses of semi-supervised learning-based models.

Scheme	Strengths	Weaknesses
[121]	Designing a semi-supervised deep learning approach, high accuracy, acceptable runtime	Insufficient experiments to evaluate the final model, not considering the effect of different parameters on the final learning model, not using different deep learning algorithms to evaluate the learning model
[122]	Evaluating the performance of the learning model based on different datasets, evaluating the performance of the learning model based on different conditions	Not considering runtime, not using different deep learning algorithms to evaluate the learning model, not considering a suitable pre-processing scheme
[123]	High accuracy, designing a classifier with different classes, using a semi-supervised learning to update the predicted label	Not considering runtime, not using different artificial neural networks to evaluate the learning model, not considering a suitable pre-processing scheme, insufficient experiments to evaluate the final model
[124]	Evaluating the learning model based on different datasets, considering various conditions to evaluate the learning model, high accuracy, accepted runtime	Not using different deep learning algorithms to evaluate the learning model, not considering a suitable pre-processing scheme, needing high time for the training process
[125]	Evaluating the learning model based on different datasets, high accuracy	Not using different basic learning algorithms to evaluate the learning model, not mentioning any reason to use SVM and KNN as basic classifiers, not designing a suitable pre-processing scheme, not describing the feature selection process

Evaluation. TE-DLSTM uses simulation-based evaluation. It is simulated in Python software. In the simulation process, the dataset is divided into two groups, including a training set (70% of data samples) and a testing set (30% of data samples). In this method, the evaluation criteria are accuracy and runtime.

5.12. SS-BLSTM

Gupta et al. [122] presented a recurrent neural network-based method called SS-BLSTM. The purpose of this semi-supervised approach is to extract mentions related to adverse drug reaction (ADR) from Twitter. In the following, we explain this method. Tables 12 and 13 represent the most important features of the SS-BLSTM method and its weaknesses and strengths, respectively.

Problem definition. Due to easy and broad access, social networks are known as a beneficial platform for sharing health information and are an appropriate option for monitoring health status. In [122], the authors try to discover mentions related to ADR from Twitter. This is very challenging because these texts are informal and brief. Many supervised learning methods are presented for this purpose. However, their performance is not desirable because enough labeled data samples are not available. Recently, new methods have used deep neural networks, especially LSTM to solve this issue. However, they need a large database for the training process to avoid overfitting. Accordingly, in [122], the authors presented a semi-supervised method, which uses both labeled and unlabeled data.

Database. In [122], the authors used the ADR dataset collected from Twitter for the supervised learning phase. This database has been obtained from 2007 to 2010. In these tweets, there are 81 drugs. The database includes 645 tweets. The unlabeled dataset is produced using Twitter's Search API. This database includes 0.1 million tweets.

Data pre-processing method. In [122], a data normalization process is performed on the dataset to remove some words, symbols, and spaces.

ML model development. SS-BLSTM has two main steps: (1) The unsupervised learning step. The main task is to extract the drug name from tweets using an unsupervised learning

scheme. For this, a bi-LSTM is trained. In this step, its weights are updated. Finally, these weights are maintained for the second step; (2) The supervised learning phase. The main task is to extract ADR from tweets using a supervised method. In this phase, the bi-LSTM model, which has been trained in the first step, is trained again to learn the labels mentioned in the tweet text.

Evaluation. SS-BLSTM uses simulation-based evaluation. It is implemented in Python software. To evaluate the performance of this method, the labeled database is divided into two sets, including training (470 tweets) and testing (170 tweets). In the evaluation process, various parameters including F1-Score, precision, and recall are used.

5.13. ECG Classification System Based on Semi-Supervised Learning

Zhai et al. [123] suggested a semi-supervised learning system to classify electrocardiogram (ECG). The purpose of the classification is to detect arrhythmia. This learning issue classifies time series signals with unbalanced classes. It has three classes: normal beats, supraventricular ectopic beats (SVEB), and ventricular ectopic beats (VEB). The purpose of this scheme is to diagnose SVEB and VEB without labeling ECG data. Note that the authors use a two-dimensional convolutional neural network (CNN) in this scheme. In the following, we describe this scheme in detail. Moreover, Tables 12 and 13 present the specifications of this system and its advantages and disadvantages, respectively.

Problem definition. Electrocardiogram (ECG) is a useful tool to detect arrhythmia. However, ECG interpretation is a very difficult, time consuming, and expert task. However, collecting ECG information is almost simple. Therefore, it is very necessary to design an automatic ECG classification system. Today, there are many techniques for classifying time series, but their performance is not acceptable. This is because enough labeled data is not available. Therefore, the combination of both unlabeled and labeled data can improve the performance of an ECG classifier. As a result, the authors of [123] select semi-supervised learning for designing such a system.

Dataset. In [123], two datasets are used for modeling this system: (1) The MIT-BIH arrhythmia database, which includes 48 ECG recorded for 47 people. In this database, each record includes ECG data for 30 min. The label of each record is determined by an expert; (2) Unlabeled database, often data samples in this database are normal beats. This allows the classifier to learn the normal beats specifications.

Data pre-processing method. In [123], a data normalization process is performed on the dataset.

ML model development. This learning model has three main steps. In the first step, an unsupervised learning process is used to accurately detect normal beats based on unlabeled ECG data. In the second step, the CNN classifier is trained using the MIT-BIH dataset and the normal beats estimated in Step 1. Then, a semi-supervised process is performed for updating labels extracted from CNN to improve its performance.

Evaluation. This method uses simulation-based evaluation. It is simulated in MATLAB software. In the evaluation process, the MIT-BIH database is divided into two parts, including the training set (22 records) and the testing set (22 records). Criteria parameters are accuracy, sensitivity, specificity, PPR, and F1-Score.

5.14. A Deep Learning Model for Segmenting Retinal Fundus Images

Bengani et al. [124] offered a deep learning model for segmenting the optic disk in retinal images. This method uses two learning techniques, including semi-supervised learning and transfer learning. In the following, we explain this method in summary. In addition, Tables 12 and 13 represent the main characteristics of this method and its advantages and disadvantages, respectively.

Problem definition. Ophthalmologists use retinal images to detect eye diseases such as retinopathy. The location connecting the optic nerve to the retina is called the optic disk (OD). Detecting the optic disk in retinal images is very challenging, time-consuming. Therefore, computer diagnostic systems are very useful tools to segment and measure OD.

The purpose of this system is to automatically detect OD for providing proper and timely treatment services. Today, deep learning models, especially artificial neural networks such as CNN have been used to do this work. These networks have a very good learning ability. However, they need a large database for training to avoid overfitting. On the other hand, the databases available for deep retinal images are very small. In [124], the authors attempt to overcome these problems using semi-supervised learning and transfer learning.

Dataset. In [124], the authors use various databases. These databases are: (1) Kaggle's diabetic retinopathy database. The authors employ this labeled dataset for training the auto-encoder network. It includes 88702 retinal images; (2) DRISHTI GS1 database. The authors use this dataset for the segmentation network. It includes 101 retinal images. The authors divide this dataset into two parts, including the training set (50 images) and the testing set (50 images); (3) RIM-ONE database. This database includes 159 retinal images. Experts segment these images and determine OD in these images. The segmentation network utilizes this dataset.

Data pre-processing method. In the first step, the auto-encoder network and the segmentation network perform a two-phase data pre-processing scheme. In the first phase, image size is changed. The purpose of this phase is to normalize images and adjust their size. In the second phase, data augmentation is performed. The purpose of this phase is to increase the number of instances. This work is performed using different transformations on the input image.

ML model development. In the first step, a deep neural network called convolutional auto-encoder (CAE) is employed. This network is trained based on the unlabeled database. The aim is to learn the features of images based on input data to rebuild output images. Then, a convolutional layer is added to this trained CAE. In this case, it is converted to the segmentation network. In this step, transfer learning is used. This means that weights are obtained according to the trained CAE model. Then, the segmentation network is again trained using the labeled dataset. Finally, this model can be used to detect OD in retinal images.

Evaluation. This method uses simulation-based evaluation. It is simulated by the TensorFlow tool in Python. The evaluation scales are DSC, Jaccard index, accuracy, sensitivity, and specificity. Note that the times required for training the CAE network and the segmentation network are 10 h and 26 min and one hour and 31 min, respectively. The times required for testing on the DRISHTI and RIM-ONE datasets are 1.19 and 1.4 s, respectively.

5.15. A Semi-Supervised Learning Method Based on GAN

Yang et al. [125] proposed a semi-supervised learning scheme, which uses the generative adversarial networks (GAN). The purpose of this scheme is to improve clinical detections in the IoT-based healthcare system. This method can solve two problems, including not availability of labeled medical data and imbalance classes. In the following, we describe this method. In addition, the most important characteristics of this method are in Table 12. We present its weaknesses and strengths in Table 13.

Problem definition. Today, the Internet of things (IoT) is changing our lifestyle in many areas, including healthcare. The IoT technology can produce a large amount of data for medical services. These data samples are used to produce a medical support system. The main task of this system is classification. Note that the performance of a classifier will be improved with increasing access to labeled data. However, this issue deals with various challenges, for example (1) IoT helps us to collect many medical data, but the labeled data samples are highly low; (2) In IoT, we deal with a problem called imbalanced data; this problem is due to high diversity in datasets. For solve these problems, one solution is to use semi-supervised learning. Therefore, in [125], a GAN-based semi-supervised learning method is presented.

Dataset. In [125], the authors utilize 10 UCI balanced datasets and 10 UCI unbalanced datasets. The number of data samples in these datasets is between 80 and 2000.

Furthermore, each data sample has between 3 and 30 features in these datasets. Additionally, the cerebral stroke database has been used to evaluate the performance of the learning method. This dataset includes 11,039 data samples. So that, each data sample has 33 features. This dataset includes both labeled data (100 data samples) and unlabeled data (10,939 data samples).

Data pre-processing method. In [125], the authors designed a data pre-processing module that modifies the dataset with the unbalanced classes. This module increases the size of a small labeled dataset using GAN. Then, a feature selection process is performed on the dataset. Note that the authors do not describe this module and the feature selection process exactly.

ML model development. In the first step, GAN receives the labeled dataset as the input to produce a number of artificial data samples. The purpose of this work is to enlarge the size of the labeled dataset and correct the unbalanced class. Then, the authors train two basic learning algorithms, including support vector machine (SVM) and K -nearest neighbors (KNN) using both the labeled dataset and artificial data samples. The purpose of these algorithms is to predict the label of unlabeled data samples. Then, the data samples with the predicted label are added to the labeled dataset. In the next step, GAN will be used again for this dataset to produce artificial data samples. The number of these artificial data samples is equal to the size of the dataset. Finally, the authors train the final classifier (i.e., SVM) using both real data samples and artificial data to perform the classification task.

Evaluation. This scheme uses simulation-based evaluation. It is implemented using MATLAB software. Note that each dataset is divided into two sections, including the training set (70% of data samples) and the testing set (30% of data samples). The evaluation scale for this method is accuracy.

5.16. Hybrid Fuzzy Clustering Scheme

Kanniappan et al. [126] segmented abnormal areas in brain MRI slides. They used fuzzy clustering to model a semi-automatic system for detecting normal and abnormal areas in each brain MRI slide. In the following, we examine this method exactly. In addition, the main specifications of this method are summarized in Table 14. Table 15 presents its strengths and weaknesses.

Problem definition. In healthcare, detecting brain tumors is a very important issue. Obtaining information about abnormal tissues is a very critical phase to detect the disease and start the treatment process. The segmentation techniques can help radiologists to discover these abnormalities in MRI. Today, computer-based methods can efficiently diagnose brain tumors. One solution for this issue is clustering. In particular, fuzzy clustering technique is a suitable method for segmenting MR images to diagnose brain tumors. Therefore, in [126], the authors presented a hybrid fuzzy clustering method to solve this issue.

Dataset. In [126], the authors used two MRI datasets: (1) A real medical dataset. It includes 22 brain slides. Proscans Diagnostics Center has produced these images; (2) BRATS dataset. It includes information about 10 individuals. In this dataset, there are 200 brain slides for each patient.

Data pre-processing method. In [126], in the first step, the authors preprocess these slides to normalize their size. So that they are represented as array, which is 512×512 pixels. In addition, all non-brain tissues are removed from MR images to improve the performance of this scheme.

Table 14. The most important characteristics of unsupervised learning-based models.

Scheme	Target	Data Preprocessing Technique	Learning Model	Evaluation Criteria	Simulator
[126]	Segmenting brain MRI and detecting brain tumors	Data normalization method	Unsupervised learning (Fuzzy clustering)	PSNR: 28.802 NCC: 0.717 NAE: 0.455 SSIM: 0.814 Jaccard: 0.79 Dice: 0.88	Python
[127]	Detecting the social anxiety disorder (SAD)	Min-Max, IBM Modelere V18.0, and detecting noisy data using SOM	Unsupervised learning and supervised learning	Accuracy: 98.67% Sensitivity: 97.14% Specificity: 10%	–
[128]	Segmenting the thyroid nodule images to detect the papillary thyroid carcinomas (PTC)	–	Unsupervised learning	SA: 99.87% CS: 99.78%	MATLAB
[129]	Designing a HAR mechanism	FT for normalizing data samples and removing noisy data, designing a feature extraction scheme based on the coder architecture and the Z-Layer method	Unsupervised learning and supervised learning	Accuracy: 97.5% MSE: 0.52% RunTime: 11.25 ns	–
[130]	Designing an imputation algorithm for estimating missing values	Handling missing values using CLUSTIMP	Unsupervised learning and supervised learning	Mammographic Mass dataset Error rate: 9–11% HCC dataset Error rate: 4–5%	Python

Table 15. The most important strengths and weaknesses of unsupervised learning-based models.

Scheme	Strengths	Weaknesses
[126]	Evaluation of the performance of the model using different experiments, using the Silhouette index to select the number of clusters, evaluate the performance of the model with its subsequent post-paste processes	Evaluation of model performance with other clustering methods, calculating the time of implementation of the proposed method, high error, non-design of a method for removing the noise of MRI images, not evaluating the speed of clustering, evaluating the model using the large database
[127]	High accuracy, designing a suitable data pre-processing method	Not evaluating the performance of the learning model using different clustering schemes, Not evaluating the performance of the learning model based on different classifiers, not calculating runtime, insufficient experiments to evaluate the final model, not testing the learning model with other available databases
[128]	Designing a scheme for removing noise, high segmentation accuracy, considering runtime, evaluating the performance of the learning model using different noises	Not designing a suitable pre-processing scheme, not testing the learning model with other available databases
[129]	Designing a suitable data pre-processing scheme, reducing computational time, reducing error rate, high accuracy	Not testing the learning model with other available databases, insufficient experiments to evaluate the final model
[130]	Designing a suitable data pre-processing scheme for estimating missing values, reducing error rate	Not testing the learning model with large datasets, not calculating runtime

ML model development. In [126], the authors used the fuzzy clustering (FC) technique to segment MR images. The purpose of fuzzy clustering is to group m data samples of the brain slide into k clusters. After the clustering process, each data sample achieves a membership degree for a specific cluster, so that the data sample closest the cluster center has the highest membership degree. Then, the cluster center is calculated based on the mean of data samples. These data samples are weighted using their membership degree. In the next step, the membership degree of each data samples is updated. This process continues until the total distance between each data sample to the cluster center is minimized or the better result is not achieved. This process segments the brain structure. Note that in the clustering process, it is very important to determine the number of clusters. In [126], this work is done using the silhouette score. In the next step, extracted structures are improved through morphological operations to determine the boundary between clusters. Finally, the authors perform some post-processing techniques to extract the desired area (i.e., tumor) from brain slides.

Evaluation. This scheme uses both simulation-based evaluation and practical implementation-based evaluation. It is implemented in Python software. Some evaluation criteria are Peak Signal to Noise Ratio (PSNR), Normalized Cross-Correlation (NCC), Normalized Absolute Error (NAE) and Structural Similarity Index (SSIM). The performance of hybrid fuzzy clustering is evaluated based on some similarity criteria such as Dice and Jaccard. Note that this method practically evaluates the brain MR images of a particular patient.

5.17. An Medical Support System for Detecting Social Anxiety Disorder

Fathi et al. [127] designed a medical support system for detecting social anxiety disorder (SAD). The authors used the self-organizing map (SOM) to detect noisy data. SAD is detected through an adaptive neuro-fuzzy inference system (ANFIS) technique. In the following, we describe this method in detail. Table 14 expresses the most important features of this method. Furthermore, Table 15 presents its advantages and disadvantages.

Problem definition. Social anxiety disorder (SAD) is one of the most common phobias. Psychiatrists face with many challenges for detecting this disease because patients do not have enough knowledge about this disorder. Therefore, it is very useful to design a medical support system for detecting SAD. In [127], ANFIS is used for modeling such a system. ANFIS is an appropriate learning model that utilizes the advantages of artificial neural networks and fuzzy logic. This means that the fuzzy system helps ANFIS to solve uncertainties and ambiguities, and the neural network helps ANFIS to manage noisy data.

Dataset. In this method, the authors achieve primary raw data through a website. The dataset includes information about 214 patients. Each data sample has 11 features. Note that the dataset has no missing values.

Data pre-processing method. In [127], the data pre-processing scheme has three steps: (1) Data normalization. The purpose of the data normalization process is that different features have the same effect on the final learning model. In [127], the authors used the Min-Max normalization method; (2) The feature selection process. The purpose of this step is to decrease the model complexity, save the time required for training model, lower data dimensionality, and avoid overfitting. The feature selection process is performed using SPSS Modeler V18.0 software to select seven useful features for detecting SAD; (3) Noise detection. In [127], SOM technique has been used for noise detection. After the clustering process, clusters that includes a small number of data samples (one or two data samples) are considered as noisy data and are removed from the dataset. Then, the cluster's behavior is evaluated based on two standards, namely social phobia inventory (SPIN) and Liebowitz social anxiety (LSA). After this evaluation, if clusters have abnormal behavior then they are recognized as noisy data. Therefore, they are removed from the dataset. After this step, 63 data samples are removed from the dataset. As a result, the dataset has 151 data samples.

ML model development. The authors of [127] used the ANFIS classifier to detect SAD disorder. It is a combination of fuzzy logic and neural network. This algorithm is trained

using least square and back-propagation methods. ANFIS has five layers. The first layer refers to input layer and final layer indicates output.

Evaluation. This method uses simulation-based evaluation. Note that the authors do not mention any description about simulator. The five-Fold Cross-Validation technique validates this scheme. Evaluation criteria include accuracy, sensitivity, and specificity.

5.18. AFGC

Huang [128] suggested an adaptive fast generalized fuzzy C-means clustering (AFGC) algorithm. The purpose of this method is to segment the thyroid nodule images in a noisy environment to accurately detect malignant thyroid tumors. In the following, we describe this method in detail. Table 14 expresses the specifications of this method in summary. Furthermore, Table 15 presents its strengths and weaknesses.

Problem definition. The most common malignant thyroid is called the papillary thyroid carcinomas (PTC), which must be treated timely to stop or control this disease. Usually, ultrasound images are applied for detecting this disease. However, interpreting these images is a very difficult, time-consuming, and expert task. Therefore, computer-based systems are very beneficial for analyzing ultrasound images. The existing clustering methods for segmenting ultrasound images have poor performance and are not sufficiently accurate. This is because these images are highly noisy. In [128], a suitable segmentation model has been proposed based on the AFGC clustering method.

Database. In [128], the authors used the Jinshan Hospital database including thyroid nodule images. The PACS system is used to take these images from January 2014 to April 2016. In general, there are 610 thyroid nodule images related to 543 patients. These images are divided into two classes, including benign (403 patients) and malignant (207 patients). This dataset is used as the training set. In addition, the testing set includes the thyroid nodule images from May 2016 to September 2016. The testing set includes information about 45 patients and 50 thyroid nodule images.

Data pre-processing method. In [128], the authors did not perform any data pre-processing scheme on the database.

ML model development. In [128], the authors presented an AFGC-based segmentation algorithm to accurately segment the thyroid nodule images. In the first step, the authors determine a balance scale. This scale is calculated based on the noise probability of non-local pixels. This work helps the scheme to determine the structure information in the image exactly. In the second step, the AFGC algorithm and the weighted image are merged together. In this process, the authors consider the balance scale. This operation produces a filtered image. This scheme performs the filtering process dynamically. This means that if this image has high noise, then this scheme increases the filtering degree. Otherwise, it reduces the degree.

Evaluation. This scheme uses simulation-based evaluation. It is simulated using MATLAB software. Two evaluation scales, including segmentation accuracy (SA) and comparison scores (CS), have been used to evaluate this method.

5.19. UDR-RC

Janarthanan et al. [129] offered the unsupervised deep learning assisted reconstructed coder (UDR-RC). The purpose of this method is to present a data pre-processing scheme to optimize the dataset. In the following, we explain this method in detail. Moreover, we represent the main specifications of the UDR-RC method in Table 14. Table 15 expresses its advantages and disadvantages.

Problem definition. Human activity recognition (HAR) has created opportunities for designing e-health methods. It uses wearable sensors to recognize different body activities. These sensors are very important for detecting different diseases and selecting a suitable treatment policy. Their output is a signal. This signal must be analyzed using deep learning approaches like DCCN. For analyzing these signals, existing models have high

computational time and a lot of error rate. This means that they are not sufficiently accurate. Therefore, in [129], the UDR-RC method is presented to solve the stated problems.

Dataset. UDR-RC employs the WISDM database. The wearable sensors sense these data samples. These data samples indicate six human activities, such as walking, running, upstairs, downstairs, sitting, and standing.

Data pre-processing method. UDR-RC is a data pre-processing method, including feature selection and feature extraction. It reduces computational time and the error rate, and enhances accuracy.

ML model development. UDR-RC is designed to extract automatically high-level features. This process includes several steps. In the first step, data samples are analyzed. The purpose of this step is to represent data samples analytically. It also reduces noise in data samples. The data samples are signals based on time and frequency. In [129], Fourier transformation (FT) is used to analyze these data samples. In this scheme, a signal with a long time is broken into smaller parts. In [129], these time series are divided using a time window with constant size. In the second step, the feature extraction is performed. This step is the core of the UDR-RC method. For this purpose, the coder architecture and the Z-Layer method are merged. They create a deep learning framework. The coder architecture is an encoder-decoder architecture, which processes the input signal to extract its features using the Z-Layer method. In the third step, UDR-RC performs a feature selection process to select the most suitable features for HAR. Finally, an artificial neural network (ANN) is used for classifying human activity. It includes an input layer, an output layer, and three hidden layers.

Evaluation. UDR-RC uses simulation-based evaluation. However, the authors do not mention the software used for implementing this method. In this scheme, evaluation scales include accuracy, MSE, and runtime.

5.20. CLUSTIMP

Shobha and Savarimuthu [130] presented a clustering-based imputation technique called CLUSTIMP. In the following, we describe this method in detail. Furthermore, Table 14 expresses the most important characteristics of the CLUSTIMP method. Table 15 presents its advantages and disadvantages.

Problem definition. Healthcare datasets have useful information. However, they often include many missing values, unbalanced classes, and other problems. Missing values are known as a serious problem in these datasets. This problem can be solved using two schemes: (1) Marginalization, In this scheme, data samples with missing values are removed from the dataset; (2) Imputation, This scheme estimates the missing values. The marginalization method causes the imbalance class problem; While the imputation method does not have this problem. Therefore, in [130], an unsupervised learning algorithm is provided for estimating these missing values.

Dataset. In [130], the authors used two databases, including the mammographic mass dataset and the HCC dataset. The mammographic mass dataset has been obtained from the UCI repository. It includes 961 data samples. These data samples have six features. There are 162 data samples with missed values. Furthermore, the HCC database includes information about 165 patients. Each data sample has 50 features. In this dataset, there are missing values (10.22% of data samples).

Data pre-processing method. CLUSTIMP is a data pre-processing scheme for estimating missing values.

ML model development. In [130], the authors presented a clustering-based imputation algorithm called CLUSTIMP. This imputation model employs ART2 for creating clusters. ART2 is an unsupervised learning algorithm, which is rooted in the ART scheme. This scheme works with continuous features. After creating the cluster, each cluster has two types of data samples, including complete data samples and data samples with missing values. In the next step, cluster members are divided into two groups, including group 1 (complete data samples) and group 2 (data samples with missing values). Then, missing

values are estimated using two methods, including Expectation Maximization (EM) and J48 (a decision tree). Note that numerical missing values are imputed using EM and categorical missing values are imputed using J48.

Evaluation. CLUSIMP uses simulation-based evaluation. It is implemented in Python 2.7 software. Evaluation criteria include error rate, accuracy, and root mean squared error (RMSE).

6. Discussion

In this section, we provide some points about the ML-based methods in healthcare according to the learning models examined in Section 5. Note that the real-world datasets in the healthcare field often deal with various problems such as missing values, noisy data, high data dimensionality (a high number of features), and among others. These problems reduce the quality of datasets. This problem negatively affects the performance of ML-based models. According to the research done in this paper, we deduce that most ML-based methods in medicine consider the data pre-processing methods. Data with missing values is the most common problem in healthcare datasets. Based on the ML-based methods studied in this paper, we find that there are two main strategies for solving this problem: (1) Deleting data with missing values; (2) Estimating missing values. Qin et al. in [101], Wang et al. in [110], Baucum et al. in [120], and Savarimuthu and Shobha in [130] offered various designs for estimating missing values. Li et al. [102], Abdar and Makarenkov [107], and Wang et al. in [110] removed data with missing values from datasets. It is a simple approach for solving this problem; however, it can lead to a new problem called imbalanced classes. This problem has a negative effect on the performance of learning models. Therefore, methods, which impute missing values, provide a more appropriate solution to solve this issue. However, when designing a method for estimating missing values, it is very important to estimate missing values exactly. Otherwise, the learning model does not have an accepted performance. Wang et al. in [110] provided a hybrid method for solving this issue. This means that some data samples with high missing values are removed from the dataset and some data samples with low missing values are also imputed. In addition, most ML-based methods consider the data normalization process. The purpose of data normalization is that variables with different scales are standardized in a certain range, for example $[0, 1]$, to have the same effect on the learning model. For example, Li et al. in [102], Baucum et al. in [120], Gupta et al. in [122], Zhai et al. in [123], Bengani et al. in [124], Kanniappan et al. in [126], Fathi et al. in [127], and Janarrhanan et al. in [129] used the data normalization methods. Noise is another problem in healthcare datasets. It reduces the accuracy of learning models and increases their error. Therefore, it is very important to design approaches to remove noisy data to improve the performance of ML-based models. Data has different types, for example digital images, numerical data, and qualitative data. The noise removal process varies according to the data type in datasets. In this paper, we examined different methods for removing different types of noise in various datasets. For example, Ma et al. in [109], Fathi et al. in [127], Huang in [128], and Janarrhanan et al. in [129] provided various approaches to remove noise from data. We examined these methods in Section 5. Another important point is that the healthcare datasets often have high dimensions. This means that data samples have many features. This can increase the model complexity and boost learning time, and lead to overfitting. To solve this problem, the appropriate solution is to use methods for reducing dimensionality such as feature selection and feature extraction. Some research works have focused on feature selection and feature extraction. For example, Qin et al. in [101], Li et al. in [102], Abdar et al. in [108], Ma et al. in [109], Tseng et al. in [112], Zhu et al. in [121], Yang et al. in [125], Fathi et al. in [127], and Janarrhanan et al. in [129] provided approaches for reducing dimensionality. However, some of the methods studied in this paper do not explain the method used for reducing dimensionality. This is an important weakness in these methods because we cannot validate the results presented in these schemes to review the effect of the feature selection method on their performance.

For example, Abdar et al. in [108] and Yang et al. in [125] did not provide any explanation about the feature selection process. Table 16 categorizes the ML-based methods based on data pre-processing methods.

Table 16. Classification of ML-based methods in terms of data pre-processing schemes.

Number	Scheme	Data Pre-Processing Schemes				
		Data Cleaning Methods			Data Reduction Methods	
		Missing Value Management	Noisy Data Management	Data Normalization	Feature Selection	Feature Extraction
1	[101]	✓	×	×	✓	×
2	[102]	✓	×	✓	✓	×
3	[107]	✓	×	×	×	×
4	[108]	×	×	×	✓	×
5	[109]	×	✓	×	✓	✓
6	[110]	✓	×	×	×	×
7	[111]	×	×	×	×	×
8	[112]	×	×	×	✓	×
9	[113]	×	×	×	×	×
10	[120]	✓	×	✓	×	×
11	[121]	×	×	×	×	✓
12	[122]	×	×	✓	×	×
13	[123]	×	×	✓	×	×
14	[124]	×	×	✓	×	×
15	[125]	×	×	×	✓	×
16	[126]	×	×	✓	×	×
17	[127]	×	✓	✓	✓	×
18	[128]	×	✓	×	×	×
19	[129]	×	✓	✓	✓	✓
20	[130]	✓	×	×	×	×

Another important point in ML-based models is the type of learning algorithm used for their development. According to our reviews in this paper, it can be found that unsupervised learning-based methods are often used for data pre-processing applications. For example, Fathi et al. in [127] used the self-organizing map (SOM) for detecting noise. Janarrhanan et al. in [129] presented an unsupervised deep learning method for feature extraction, feature selection, and noise removal to reduce computational time. Savarimuthu and Shoha in [130] provided an unsupervised neural network for estimating missing values in the dataset. While supervised learning methods are often used to diagnose and classify a disease. For example, the learning approaches provided by Qin et al. [101], Li et al. [102], Abdar and Makarenkov [107], Abdar et al. [108], Ma et al. [109]. Today, deep learning methods are also used to design treatment recommendation systems. However, an important problem in these methods is that their performance depends on the labeled database. A supervised learning algorithm has good performance when enough labeled data are available for training and testing this model. However, in the healthcare field, we often do not access large labeled datasets. This can lead to an overfitting problem. This reduces the generalizability of the learning model and increases its error. Furthermore, some authors have provided solutions to solve this issue. One solution to such a problem is to use rein-

forcement learning. For example, Wang et al. in [110], Dia et al. in [111], Tseng et al. in [112], Khalilpourazari and Hashemi in [113], and Baucum et al. in [120] employed reinforcement learning for designing the learning models. However, the most important problem when using this technique in healthcare is that a reinforcement learning method should track the patient’s health status continuously to learn the optimal treatment strategy. According to the text presented above, firstly, a very difficult work is to track the patient’s health status. Secondly, researchers cannot do unauthorized tests on the patient’s body. A solution for these problems is to create an artificial environment for reinforced learning-based models. For example, Dia et al. in [111], Tseng et al. in [112], and Baucum et al. in [120] designed an artificial environment using deep learning techniques to interact with reinforcement learning-based models. Another solution to solve data unavailability is to produce artificial data samples. For example, Tseng et al. in [112] and Yang et al. [125] used a deep neural network called GAN to produce artificial data samples and enlarge the initial dataset. Another solution for data unavailability is to use semi-supervised learning methods. These methods use a combination of labeled data and unlabeled data for designing the learning model. Moreover, these methods use both learning techniques, including supervised learning and unsupervised learning. For example, Zhu et al. in [121], Gupta et al. in [122], Zhai et al. in [123], Bengani et al. in [124], and Yang et al. in [125] used semi-supervised learning for designing the learning model. Table 17 categorizes the ML-based methods in the healthcare field in terms of various learning techniques.

Table 17. Classification of ML-based methods in terms of various learning algorithms.

Number	Scheme	Various Learning Methods			
		Unsupervised Learning	Supervised Learning	Semi-Supervised Learning	Reinforcement Learning
1	[101]	×	✓	×	×
2	[102]	×	✓	×	×
3	[107]	×	✓	×	×
4	[108]	×	✓	×	×
5	[109]	×	✓	×	×
6	[110]	×	×	×	✓
7	[111]	×	✓	×	✓
8	[112]	×	✓	×	✓
9	[113]	×	×	×	✓
10	[120]	×	✓	×	✓
11	[121]	×	✓	✓	×
12	[122]	×	✓	✓	×
13	[123]	×	✓	✓	×
14	[124]	×	✓	✓	×
15	[125]	×	×	✓	×
16	[126]	✓	×	×	×
17	[127]	✓	✓	×	×
18	[128]	✓	×	×	×
19	[129]	✓	✓	×	×
20	[130]	✓	✓	×	×

When examining the ML-based methods in healthcare, another point is that researchers often evaluate the performance of their learning model using simulation software. However, this evaluation method is very important, but we believe that it is not enough. Because the ML-based methods in healthcare should be analyzed in real environments and are evaluated by physicians and specialists in this area to identify their weaknesses. In the research done in this paper, only Wang et al. in [110] and Kanniappan et al. in [126] examined their methods in a real environment, but it is highly limited. Note that the practical implementation of the learning models in healthcare is very costly. They deal with hardware complexities for implementing the ML-based models. Additionally, it is very difficult to repeat different scenarios. These problems are often considered as important obstacles for artificial intelligence researchers because they need to evaluate their own models to update them continuously. In Table 18, the ML-based methods in healthcare are categorized in terms of evaluation methods.

Table 18. Classification of ML-based methods in terms of evaluation methods.

Number	Scheme	Evaluation Methods	
		Simulation-Based Evaluation	Practical Implementation-Based Evaluation
1	[101]	✓	×
2	[102]	✓	×
3	[107]	✓	×
4	[108]	✓	×
5	[109]	✓	×
6	[110]	✓	✓
7	[111]	✓	×
8	[112]	✓	×
9	[113]	✓	×
10	[120]	✓	×
11	[121]	✓	×
12	[122]	✓	×
13	[123]	✓	×
14	[124]	✓	×
15	[125]	✓	×
16	[126]	✓	✓
17	[127]	✓	×
18	[128]	✓	×
19	[129]	✓	×
20	[130]	✓	×

The final point on the ML-based models in the healthcare field is that most ML-based methods are used to diagnose a disease. The number of papers presented in the treatment field, which use machine learning techniques is very limited. Therefore, researchers must work in this area to resolve its problems. For example, Wang et al. in [110], Dai et al. in [111], Tseng et al. [112], and Baucum et al. in [120]. Table 19 compares the ML-based methods in healthcare are in terms of application.

Table 19. Classification of ML-based methods in terms of various applications.

Number	Scheme	Application	
		Diagnosis	Treatment
1	[101]	✓	×
2	[102]	✓	×
3	[107]	✓	×
4	[108]	✓	×
5	[109]	✓	×
6	[110]	×	✓
7	[111]	✓	✓
8	[112]	×	✓
9	[113]	✓	×
10	[120]	×	✓
11	[121]	✓	×
12	[122]	✓	×
13	[123]	✓	×
14	[124]	✓	×
15	[125]	✓	×
16	[126]	✓	×
17	[127]	✓	×
18	[128]	✓	×
19	[129]	✓	×
20	[130]	✓	×

The state-of-the-art survey had presented a comprehensive review of the applications of machine learning in medical sciences. From cardiovascular disease [131], to pandemic research [132], various methods had been considered and notable methods presented. Machine learning in particular showed an exponential increase in COVID-19 research where novel methods proposed [133–140]. It has been shown that ensemble, deep learning, and hybrid methods are rapidly getting popularity as also stated in previous surveys, for example, [140–146]. The progress on the applications of evolutionary methods, for example, [147–150] in training the machine learning methods had not been progressive as other fields.

7. Challenges and Open Issues

In this section, we present some challenges and limitations when designing ML-based methods.

- Data availability: ML-based models often require large databases for training. When datasets are large, the performance of these models is well and their error is low. For this purpose, it is necessary to design new methods to record electronically medical data to solve this problem.
- Data quality: Another important point is that any unintentional or intentional error during recording data increases error rate. Therefore, data quality is a very important issue. These problems can occur when physicians and specialists are not careful enough when determining the label of data samples. Data preprocessing methods can significantly reduce these problems and improve the quality of the datasets.

- High dimensions: The real-time healthcare datasets have high dimensional. This problem increases the model complexity, boosts the learning time, and leads to overfitting. Therefore, ML-based methods should always consider this issue. There are some effective techniques for reducing dimensionality. For example, feature selection and feature extraction are effective solutions for solving this problem. However, this area requires more research to provide more efficient methods for reducing dimensionality.
- Efficiency: ML-based models are beneficial in healthcare when they solve a serious problem in this area. In some cases, we do not need to use machine learning techniques for solving a problem and these techniques are not really necessary, and existing methods can successfully resolve the problem. ML-based methods are necessary when datasets have high dimensional or all parameters are not easily predictable, or we require a long time to infer the correct results, or ordinary methods are inefficient for solving this issue. Therefore, researchers must use timely and truly machine learning techniques.
- Privacy: When designing ML-base models, we must consider the privacy issue, because patients may be identified based on anonymous data. Privacy of patients is a very important and vital problem that should be considered by researchers to do more research for addressing this problem.

8. Conclusions

In this paper, we examined ML-based methods in healthcare. For this purpose, we first explained machine learning in summary and we expressed its application in healthcare. Then, we introduced a general framework for designing ML-based models in medicine. We classified ML-based methods in medicine based on data pre-processing methods (data cleaning methods, data reduction methods), learning methods (unsupervised learning, supervised learning, semi-supervised learning, and reinforcement learning), evaluation methods (simulation-based evaluation and practical implementation-based evaluation in real environment), and applications (diagnosis, treatment). Finally, we studied some ML-based methods in healthcare and expressed their strengths and weaknesses. In this paper, we seek to provide researchers with a good view of the use of machine learning in healthcare and familiarize them with the newest research on ML applications in medicine so that they can provide new solutions to the existing problems in this area. In the future, we try to focus on deep learning techniques and reinforcement learning techniques because these techniques are very powerful tools for solving problems in healthcare.

Author Contributions: Conceptualization, M.S.Y. and E.Y.; methodology, M.S.Y., E.Y. and M.H.; validation, A.M.R., A.H. and Z.M.; investigation, A.M.R., A.H. and R.A.N.; resources, A.M.R., A.H. and Z.M.; writing—original draft preparation, M.S.Y., E.Y. and M.H.; supervision, M.H.; project administration, R.A.N. and M.H.; funding acquisition, R.A.N. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Schuld, M.; Sinayskiy, I.; Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **2015**, *56*, 172–185. [[CrossRef](#)]
2. Char, D.S.; Abràmoff, M.D.; Feudtner, C. Identifying ethical considerations for machine learning healthcare applications. *Am. J. Bioeth.* **2020**, *20*, 7–17. [[CrossRef](#)] [[PubMed](#)]
3. Nordlinger, B.; Villani, C.; Rus, D. *Healthcare and Artificial Intelligence*; Springer Nature: Cham, Switzerland, 2020. [[CrossRef](#)]
4. Johri, S.; Goyal, M.; Jain, S.; Baranwal, M.; Kumar, V.; Upadhyay, R. A novel machine learning-based analytical framework for automatic detection of COVID-19 using chest X-ray images. *Int. J. Imaging Syst. Technol.* **2021**, *31*, 1105–1119. [[CrossRef](#)]

5. Pattnayak, P.; Jena, O.P. Innovation on Machine Learning in Healthcare Services—An Introduction. *Mach. Learn. Healthc. Appl.* **2021**, 1–15. [[CrossRef](#)]
6. Reig, B.; Heacock, L.; Geras, K.J.; Moy, L. Machine learning in breast mri. *J. Magn. Reson. Imaging* **2020**, *52*, 998–1018. [[CrossRef](#)] [[PubMed](#)]
7. Demirhan, A. Neuroimage-based clinical prediction using machine learning tools. *Int. J. Imaging Syst. Technol.* **2017**, *27*, 89–97. [[CrossRef](#)]
8. Datta, S.; Barua, R.; Das, J. Application of artificial intelligence in modern healthcare system. In *Alginates recent Uses of This Natural Polymer*; IntechOpen: Rijeka, Croatia, 2020. [[CrossRef](#)]
9. Elsebakhi, E.; Lee, F.; Schendel, E.; Haque, A.; Kathireason, N.; Pathare, T.; Syed, N.; Al-Ali, R. Large-scale machine learning based on functional networks for biomedical big data with high performance computing platforms. *J. Comput. Sci.* **2015**, *11*, 69–81. [[CrossRef](#)]
10. Bashir, S.; Qamar, U.; Khan, F.H.; Naseem, L. Hmv: A medical decision support framework using multi-layer classifiers for disease prediction. *J. Comput. Sci.* **2016**, *13*, 10–25. [[CrossRef](#)]
11. Waring, J.; Lindvall, C.; Umeton, R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artif. Intell. Med.* **2020**, *104*, 101822. [[CrossRef](#)]
12. Coronato, A.; Naeem, M.; Pietro, G.D.; Paragliola, G. Reinforcement learning for intelligent healthcare applications: A survey. *Artif. Intell. Med.* **2020**, *109*, 101964. [[CrossRef](#)]
13. Yousefpoor, E.; Barati, H.; Barati, A. A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2021**, 1–26. [[CrossRef](#)]
14. Yousefpoor, M.S.; Yousefpoor, E.; Barati, H.; Barati, A.; Movaghar, A.; Hosseinzadeh, M. Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. *J. Netw. Comput. Appl.* **2021**, 103118. [[CrossRef](#)]
15. Rong, G.; Mendez, A.; Assi, E.B.; Zhao, B.; Sawan, M. Artificial intelligence in healthcare: Review and prediction case studies. *Engineering* **2020**, *6*, 291–301. [[CrossRef](#)]
16. Seaton, H. *The Construction Technology Handbook*; John Wiley & Sons: Hoboken, NJ, USA, **2021**; ISBN 978-1-119-71995-3.
17. Chen, Y.L.; Guo, Q.D. Emerging coronaviruses: Genome structure, replication, parthenogenesis. *J. Virol.* **2020**, *92*, 418423. [[CrossRef](#)]
18. Mohammed, M.; Khan, M.B.; Bashier, E.B.M. *Machine Learning: Algorithms and Applications*; CRC Press: Boca Raton, FL, USA, **2016**; ISBN 978-1-4987-0538-7.
19. Seo, H.; Khuzani, M.B.; Vasudevan, V.; Huang, C.; Ren, H.; Xiao, R.; Jia, X.; Xing, L. Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications. *Med. Phys.* **2020**, *47*, e148–e167. [[CrossRef](#)] [[PubMed](#)]
20. Zhang, X.-D. Machine Learning. In *A Matrix Algebra Approach to Artificial Intelligence*; Springer: Singapore, **2020**; pp. 223–440. [[CrossRef](#)]
21. Chen, P.-H.C.; Liu, Y.; Peng, L. How to develop machine learning models for healthcare. *Nat. Mater.* **2019**, *18*, 410–414. [[CrossRef](#)]
22. He, J.; Baxter, S.L.; Xu, J.; Xu, J.; Zhou, X.; Zhang, K. The practical implementation of artificial intelligence technologies in medicine. *Nat. Med.* **2019**, *25*, 30–36. [[CrossRef](#)]
23. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [[CrossRef](#)]
24. Uprety, A.; Rawat, D.B. Reinforcement learning for iot security: A comprehensive survey. *IEEE Internet Things J.* **2020**, *8*, 8693–8706. [[CrossRef](#)]
25. Yu, K.-H.; Beam, A.L.; Kohane, I.S. Artificial intelligence in healthcare. *Nat. Biomed. Eng.* **2018**, *2*, 719–731. [[CrossRef](#)]
26. Yousefpoor, M.S.; Barati, H. Dskms: A dynamic smart key management system based on fuzzy logic in wireless sensor networks. *Wirel. Netw.* **2020**, *26*, 2515–2535. [[CrossRef](#)]
27. Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; Dudley, J.T. Deep learning for healthcare: Review, opportunities and challenges. *Brief. Bioinform.* **2018**, *19*, 1236–1246. [[CrossRef](#)] [[PubMed](#)]
28. Alafif, T.; Tehame, A.M.; Bajaba, S.; Barnawi, A.; Zia, S. Machine and deep learning towards covid-19 diagnosis and treatment: Survey, challenges, and future directions. *Int. J. Environ. Res. Public Health* **2021**, *18*, 1117. [[CrossRef](#)]
29. Tayarani-N, M.-H. Applications of artificial intelligence in battling against covid-19: A literature review. *Chaos Solitons Fractals* **2020**, 110338. [[CrossRef](#)] [[PubMed](#)]
30. Smiti, A. When machine learning meets medical world: Current status and future challenges. *Comput. Sci. Rev.* **2020**, *37*, 100280. [[CrossRef](#)]
31. Shouval, R.; Fein, J.A.; Savani, B.; Mohty, M.; Nagler, A. Machine learning and artificial intelligence in haematology. *Br. J. Haematol.* **2021**, *192*, 239–250. [[CrossRef](#)]
32. Olsen, C.R.; Mentz, R.J.; Anstrom, K.J.; Page, D.; Patel, P.A. Clinical applications of machine learning in the diagnosis, classification, and prediction of heart failure. *Am. Heart J.* **2020**, *229*, 1–17. [[CrossRef](#)]
33. Berry, M.W.; Mohamed, A.; Yap, B.W. *Supervised and Unsupervised Learning for Data Science*; Springer: Cham, Switzerland, **2019**. [[CrossRef](#)]

34. Mabrouk, E.; Ayman, A.; Raslan, Y.; Hedar, A.R. Immune system programming for medical image segmentation. *J. Comput. Sci.* **2019**, *31*, 111–125. [[CrossRef](#)]
35. Forsch, N.; Govil, S.; Perry, J.C.; Hegde, S.; Young, A.A.; Omens, J.H.; McCulloch, A.D. Computational analysis of cardiac structure and function in congenital heart disease: Translating discoveries to clinical strategies. *J. Comput. Sci.* **2021**, *52*, 101211. [[CrossRef](#)]
36. Surendar, P. Diagnosis of lung cancer using hybrid deep neural network with adaptive sine cosine crow search algorithm. *J. Comput. Sci.* **2021**, *53*, 101374. [[CrossRef](#)]
37. Saxena, A.; Chandra, S. *Artificial Intelligence and Machine Learning in Healthcare*; Springer: Singapore, **2021**. [[CrossRef](#)]
38. Pucchio, A.; Eisenhauer, E.A.; Moraes, F.Y. Medical students need artificial intelligence and machine learning training. *Nat. Biotechnol.* **2021**, *39*, 388–389. [[CrossRef](#)] [[PubMed](#)]
39. Samuel, A.L. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.* **1959**, *3*, 210–229. [[CrossRef](#)]
40. Alpaydin, E. *Introduction to Machine Learning*, 3rd ed.; PHI Publisher: New Delhi, India, 2014.
41. Kubat, M. *An Introduction to Machine Learning*; Springer: Cham, Switzerland, **2017**. [[CrossRef](#)]
42. Belciug, S.; Gorunescu, F. Era of intelligent systems in healthcare. In *Intelligent Decision Support Systems—A Journey to Smarter Healthcare*; Springer: Cham, Switzerland, **2020**; pp. 1–55. [[CrossRef](#)]
43. El Naqa, I.; Murphy, M.J. What is machine learning? In *Machine Learning in Radiation Oncology*; Springer: Cham, Switzerland, **2015**; pp. 3–11. [[CrossRef](#)]
44. Dulhare, U.N.; Ahmad, K.; Ahmad, K.A.B. (Eds.) *Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
45. Shobha, G.; Rangaswamy, S. Machine learning. In *Handbook of Statistics*; Elsevier: Amsterdam, The Netherlands, **2018**; Volume 38, pp. 197–228. [[CrossRef](#)]
46. Alsuliman, T.; Humaidan, D.; Sliman, L. Machine learning and artificial intelligence in the service of medicine: Necessity or potentiality? *Curr. Res. Transl. Med.* **2020**, *68*, 245–251. [[CrossRef](#)]
47. Kandhway, P.; Bhandari, A.K.; Singh, A. A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization. *Biomed. Signal Process. Control* **2020**, *56*, 101677. [[CrossRef](#)]
48. Zerouaoui, H.; Idri, A. Reviewing machine learning and image processing based decision-making systems for breast cancer imaging. *J. Med. Syst.* **2021**, *45*, 1–20. [[CrossRef](#)] [[PubMed](#)]
49. Handelman, G.; Kok, H.; Chandra, R.; Razavi, A.; Lee, M.; Asadi, H. Machine learning and the future of medicine. *J. Intern. Med.* **2018**, *284*, 603–619. [[CrossRef](#)] [[PubMed](#)]
50. Vatandsoost, M.; Litkouhi, S. The future of healthcare facilities: How technology and medical advances may shape hospitals of the future. *Hosp. Pract. Res.* **2019**, *4*, 1–11. [[CrossRef](#)]
51. Himidan, S.; Kim, P. The evolving identity, capacity, and capability of the future surgeon. In *Seminars in Pediatric Surgery*; WB Saunders, Elsevier: Amsterdam, The Netherlands, 2015; Volume 24, pp. 145–149. [[CrossRef](#)]
52. Assaf, D.; Rayman, S.; Segev, L.; Neuman, Y.; Zippel, D.; Goitein, D. Improving pre-bariatric surgery diagnosis of hiatal hernia using machine learning models. *Minim. Invasive Ther. Allied Technol.* **2021**, 1–7. [[CrossRef](#)] [[PubMed](#)]
53. De Bruyne, S.; Speeckaert, M.M.; Van Biesen, W.; Delanghe, J.R. Recent evolutions of machine learning applications in clinical laboratory medicine. *Crit. Rev. Clin. Lab. Sci.* **2021**, *58*, 131–152. [[CrossRef](#)] [[PubMed](#)]
54. Rahmani, A.M.; Ali, S.; Yousefpoor, M.S.; Yousefpoor, E.; Naqvi, R.A.; Siddique, K.; Hosseinzadeh, M. An area coverage scheme based on fuzzy logic and shuffled frog-leaping algorithm (sfla) in heterogeneous wireless sensor networks. *Mathematics* **2021**, *9*, 2251. [[CrossRef](#)]
55. Lee, S.-W.; Ali, S.; Yousefpoor, M.S.; Yousefpoor, E.; Lalbakhsh, P.; Javaheri, D.; Rahmani, A.M.; Hosseinzadeh, M. An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (fanets). *IEEE Access* **2021**, *9*, 129977–130005. [[CrossRef](#)]
56. Tao, W.; Concepcion, A.N.; Vianen, M.; Marijnissen, A.C.; Lafeber, F.P.; Radstake, T.R.; Pandit, A. Multiomics and machine learning accurately predict clinical response to adalimumab and etanercept therapy in patients with rheumatoid arthritis. *Arthritis Rheumatol.* **2021**, *73*, 212–222. [[CrossRef](#)]
57. Alizadehsani, R.; Roshanzamir, M.; Abdar, M.; Beykikhoshk, A.; Khosravi, A.; Panahiazar, M.; Koohestani, A.; Khozeimeh, F.; Nahavandi, S.; Sarrafzadegan, N. A database for using machine learning and data mining techniques for coronary artery disease diagnosis. *Sci. Data* **2019**, *6*, 1–13. [[CrossRef](#)]
58. Ben-Israel, D.; Jacobs, W.B.; Casha, S.; Lang, S.; Ryu, W.H.A.; de Lotbiniere-Bassett, M.; Cadotte, D.W. The impact of machine learning on patient care: A systematic review. *Artif. Intell. Med.* **2020**, *103*, 101785. [[CrossRef](#)]
59. Yousefpoor, M.S.; Barati, H. Dynamic key management algorithms in wireless sensor networks: A survey. *Comput. Commun.* **2019**, *134*, 52–69. [[CrossRef](#)]
60. Golsorkhtabar, M.; Nia, F.K.; Hosseinzadeh, M.; Vejdandparast, Y. The novel energy adaptive protocol for heterogeneous wireless sensor networks. In *Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology*, Chengdu, China, 9–11 July 2010; Volume 2, pp. 178–182. [[CrossRef](#)]
61. Nikravan, M.; Movaghar, A.; Hosseinzadeh, M. A lightweight defense approach to mitigate version number and rank attacks in low-power and lossy networks. *Wirel. Pers. Commun.* **2018**, *99*, 1035–1059. [[CrossRef](#)]

62. Zitnik, M.; Nguyen, F.; Wang, B.; Leskovec, J.; Goldenberg, A.; Hoffman, M.M. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Inf. Fusion* **2019**, *50*, 71–91. [CrossRef]
63. Stetco, A.; Dinmohammadi, F.; Zhao, X.; Robu, V.; Flynn, D.; Barnes, M.; Keane, J.; Nenadic, G. Machine learning methods for wind turbine condition monitoring: A review. *Renew. Energy* **2019**, *133*, 620–635. [CrossRef]
64. Dhal, P.; Azad, C. A comprehensive survey on feature selection in the various fields of machine learning. *Appl. Intell.* **2021**, 1–39. [CrossRef]
65. Tiwari, S.R.; Rana, K.K. Feature selection in big data: Trends and challenges. In *Data Science and Intelligent Applications*; Springer: Singapore, 2021; pp. 83–98. [CrossRef]
66. Guyon, I.; Elisseeff, A. An introduction to feature extraction. In *Feature Extraction*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–25. [CrossRef]
67. Xiong, Z.; Cui, Y.; Liu, Z.; Zhao, Y.; Hu, M.; Hu, J. Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation. *Comput. Mater. Sci.* **2020**, *171*, 109203. [CrossRef]
68. Xu, Z.; Qin, W.; Tang, Q.; Jiang, D. Energy-efficient cognitive access approach to convergence communications. *Sci. China Inf. Sci.* **2014**, *57*, 1–12. [CrossRef]
69. Mandal, I. Machine learning algorithms for the creation of clinical healthcare enterprise systems. *Enterp. Inf. Syst.* **2017**, *11*, 1374–1400. [CrossRef]
70. Feldman, K.; Faust, L.; Wu, X.; Huang, C.; Chawla, N.V. Beyond volume: The impact of complex healthcare data on the machine learning pipeline. In *Towards Integrative Machine Learning and Knowledge Extraction*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 150–169. [CrossRef]
71. Zhang, J.M.; Harman, M.; Ma, L.; Liu, Y. Machine learning testing: Survey, landscapes and horizons. *IEEE Trans. Softw. Eng.* **2020**. [CrossRef]
72. Javaheri, D.; Hosseinzadeh, M.; Rahmani, A.M. Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. *IEEE Access* **2018**, *6*, 78321–78332. [CrossRef]
73. Mesbahi, M.R.; Rahmani, A.M.; Hosseinzadeh, M. Highly reliable architecture using the 80/20 rule in cloud computing datacenters. *Future Gener. Comput. Syst.* **2017**, *77*, 77–86. [CrossRef]
74. Wu, H.; Meng, F.J. April. Review on Evaluation Criteria of Machine Learning Based on Big Data. *J. Phys. Conf. Ser.* **2020**, *1486*, 052026. [CrossRef]
75. Vamplew, P.; Dazeley, R.; Berry, A.; Issabekov, R.; Dekker, E. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Mach. Learn.* **2011**, *84*, 51–80. [CrossRef]
76. Setiawan, A.W. November. Image Segmentation Metrics in Skin Lesion: Accuracy, Sensitivity, Specificity, Dice Coefficient, Jaccard Index, and Matthews Correlation Coefficient. In Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 17–18 November 2020; pp. 97–102. [CrossRef]
77. Zhang, J.; Barr, E.; Guedj, B.; Harman, M.; Shawe-Taylor, J. Perturbed Model Validation: A New Framework to Validate Model Relevance. 2019. Available online: <https://hal.inria.fr/hal-02139208> (accessed on 24 August 2021).
78. Werpachowski, R.; György, A.; Szepesvári, C. Detecting overfitting via adversarial examples. *arXiv* **2019**, arXiv:1903.02380.
79. Molnar, C. Interpretable Machine Learning. Available online: <https://christophm.github.io/interpretable-ml-book> (accessed on 11 September 2021).
80. Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* **2019**, *267*, 1–38. [CrossRef]
81. Doshi-Velez, F.; Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv* **2017**, arXiv:1702.08608.
82. Slack, D.; Friedler, S.A.; Scheidegger, C.; Roy, C.D. Assessing the local interpretability of machine learning models. *arXiv* **2019**, arXiv:1902.03501.
83. Zhou, Z.Q.; Sun, L.; Chen, T.Y.; Towey, D. Metamorphic relations for enhancing system understanding and use. *IEEE Trans. Softw. Eng.* **2018**, *46*, 1120–1154. [CrossRef]
84. Chen, W.; Sahiner, B.; Samuelson, F.; Pezeshk, A.; Petrick, N. Calibration of medical diagnostic classifier scores to the probability of disease. *Stat. Methods Med Res.* **2018**, *27*, 1394–1409. [CrossRef] [PubMed]
85. Caruana, R.; Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 161–168. [CrossRef]
86. Alias Balamurugan, A.; Rajaram, R.; Pramala, S.; Rajalakshmi, S.; Jeyendran, C.; Prakash, J.D.S. Nb+: An improved naive bayesian algorithm. *Knowl.-Based Syst.* **2011**, *24*, 563–569. [CrossRef]
87. Ballard, Z.; Brown, C.; Madni, A.M.; Ozcan, A. Machine learning and computation-enabled intelligent sensor design. *Nat. Mach. Intell.* **2021**, *3*, 556–565. [CrossRef]
88. Miorelli, R.; Kulakovskiy, A.; Chapuis, B.; Dalmeida, O.; Mesnil, O. Supervised learning strategy for classification and regression tasks applied to aeronautical structural health monitoring problems. *Ultrasonics* **2021**, *113*, 106372. [CrossRef] [PubMed]
89. Dhasaradhan, K.; Jaichandran, R.; Shunmuganathan, K.; Kiruthika, S.U.; Rajaprakash, S. Hybrid machine learning model using decision tree and support vector machine for diabetes identification. In *Data Engineering and Intelligent Computing*; Springer: Singapore, 2021; pp. 293–305. [CrossRef]
90. Shrestha, Y.R.; Krishna, V.; von Krogh, G. Augmenting organizational decision-making with deep learning algorithms: Principles, promises, and challenges. *J. Bus. Res.* **2021**, *123*, 588–603. [CrossRef]

91. Villarrubia, G.; Paz, J.F.D.; Chamoso, P.; la Prieta, F.D. Artificial neural networks used in optimization problems. *Neurocomputing* **2018**, *272*, 10–16. [[CrossRef](#)]
92. Hasan, K.Z.; Hasan, M.Z. Performance evaluation of ensemble-based machine learning techniques for prediction of chronic kidney disease. In *Emerging Research in Computing, Information, Communication and Applications*; Springer: Singapore, **2019**; pp. 415–426. [[CrossRef](#)]
93. Gottwald, G.A.; Reich, S. Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. *Phys. D Nonlinear Phenom.* **2021**, *423*, 132911. [[CrossRef](#)]
94. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
95. Piccialli, F.; Di Somma, V.; Giampaolo, F.; Cuomo, S.; Fortino, G. A survey on deep learning in medicine: Why, how and when? *Inf. Fusion* **2021**, *66*, 111–137. [[CrossRef](#)]
96. Sharma, S.; Singh, G.; Sharma, M. A comprehensive review and analysis of supervised-learning and soft computing techniques for stress diagnosis in humans. *Comput. Biol. Med.* **2021**, 104450. [[CrossRef](#)] [[PubMed](#)]
97. Celebi, M.E.; Aydin, K. *Unsupervised Learning Algorithms*; Springer: Berlin/Heidelberg, Germany, **2016**.
98. Zhang, L.; Liu, P.; Zhao, L.; Wang, G.; Zhang, W.; Liu, J. Air quality predictions with a semi-supervised bidirectional lstm neural network. *Atmos. Pollut. Res.* **2021**, *12*, 328–339. [[CrossRef](#)]
99. Bull, L.; Worden, K.; Dervilis, N. Towards semi-supervised and probabilistic classification in structural health monitoring. *Mech. Syst. Signal Process.* **2020**, *140*, 106653. [[CrossRef](#)]
100. Xu, X.; Zuo, L.; Huang, Z. Reinforcement learning algorithms with function approximation: Recent advances and applications. *Inf. Sci.* **2014**, *261*, 1–31. [[CrossRef](#)]
101. Qin, J.; Chen, L.; Liu, Y.; Liu, C.; Feng, C.; Chen, B. A machine learning methodology for diagnosing chronic kidney disease. *IEEE Access* **2019**, *8*, 20991–21002. [[CrossRef](#)]
102. Li, J.P.; Haq, A.U.; Din, S.U.; Khan, J.; Khan, A.; Saboor, A. Heart disease identification method using machine learning classification in e-healthcare. *IEEE Access* **2020**, *8*, 107562–107582. [[CrossRef](#)]
103. Urbanowicz, R.J.; Meeker, M.; Cava, W.L.; Olson, R.S.; Moore, J.H. Relief-based feature selection: Introduction and review. *J. Biomed. Inform.* **2018**, *85*, 189–203. [[CrossRef](#)] [[PubMed](#)]
104. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, maxrelevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [[CrossRef](#)] [[PubMed](#)]
105. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **1996**, *58*, 267–288. [[CrossRef](#)]
106. Sun, Y.; Todorovic, S.; Goodison, S. Local-learning-based feature selection for high-dimensional data analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1610–1626. [[CrossRef](#)]
107. Abdar, M.; Makarenkov, V. Cwv-bann-svm ensemble learning classifier for an accurate diagnosis of breast cancer. *Measurement* **2019**, *146*, 557–570. [[CrossRef](#)]
108. Abdar, M.; Zomorodi-Moghadam, M.; Zhou, X.; Gururajan, R.; Tao, X.; Barua, P.D.; Gururajan, R. A new nested ensemble technique for automated diagnosis of breast cancer. *Pattern Recognit. Lett.* **2020**, *132*, 123–131. [[CrossRef](#)]
109. Ma, F.; Sun, T.; Liu, L.; Jing, H. Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network. *Future Gener. Comput. Syst.* **2020**, *111*, 17–26. [[CrossRef](#)]
110. Wang, L.; Zhang, W.; He, X.; Zha, H. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 19–23 August 2018; pp. 2447–2456. [[CrossRef](#)]
111. Dai, Y.; Wang, G.; Muhammad, K.; Liu, S. A closed-loop healthcare processing approach based on deep reinforcement learning. *Multimed. Tools Appl.* **2020**, 1–23. [[CrossRef](#)]
112. Tseng, H.-H.; Luo, Y.; Cui, S.; Chien, J.-T.; Haken, R.K.T.; Naqa, I.E. Deep reinforcement learning for automated radiation adaptation in lung cancer. *Med. Phys.* **2017**, *44*, 6690–6705. [[CrossRef](#)]
113. Khalilpourazari, S.; Doulabi, H.H. Designing a hybrid reinforcement learning based algorithm with application in prediction of the covid-19 pandemic in quebec. *Ann. Oper. Res.* **2021**, 1–45. [[CrossRef](#)]
114. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
115. Mirjalili, S. Sca: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
116. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [[CrossRef](#)]
117. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
118. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110*, 151–166. [[CrossRef](#)]
119. Salimi, H. Stochastic fractal search: A powerful metaheuristic algorithm. *Knowl.-Based Syst.* **2015**, *75*, 1–18. [[CrossRef](#)]
120. Baucum, M.; Khojandi, A.; Vasudevan, R. Improving deep reinforcement learning with transitional variational autoencoders: A healthcare application. *IEEE J. Biomed. Health Inform.* **2020**, *25*, 2273–2280. [[CrossRef](#)]
121. Zhu, Q.; Chen, Z.; Soh, Y.C. A novel semisupervised deep learning method for human activity recognition. *IEEE Trans. Ind. Inform.* **2018**, *15*, 3821–3830. [[CrossRef](#)]

122. Gupta, S.; Pawar, S.; Ramrakhiyani, N.; Palshikar, G.K.; Varma, V. Semi-supervised recurrent neural network for adverse drug reaction mention extraction. *BMC Bioinform.* **2018**, *19*, 1–7. [[CrossRef](#)] [[PubMed](#)]
123. Zhai, X.; Zhou, Z.; Tin, C. Semi-supervised learning for ecg classification without patient-specific labeled data. *Expert Syst. Appl.* **2020**, *158*, 113411. [[CrossRef](#)]
124. Bengani, S.; Jothi, A.A.; Vadivel, S. Automatic segmentation of optic disc in retinal fundus images using semi-supervised deep learning. *Multimed. Tools Appl.* **2021**, *80*, 3443–3468. [[CrossRef](#)]
125. Yang, Y.; Nan, F.; Yang, P.; Meng, Q.; Xie, Y.; Zhang, D.; Muhammad, K. Gan-based semi-supervised learning approach for clinical decision support in health-iot platform. *IEEE Access* **2019**, *7*, 8048–8057. [[CrossRef](#)]
126. Kanniappan, S.; Samiayya, D.; Vincent, D.R.; Srinivasan, P.M.K.; Jayakody, D.N.K.; Reina, D.G.; Inoue, A. An efficient hybrid fuzzy-clustering driven 3d-modeling of magnetic resonance imagery for enhanced brain tumor diagnosis. *Electronics* **2020**, *9*, 475. [[CrossRef](#)]
127. Fathi, S.; Ahmadi, M.; Birashk, B.; Dehnad, A. Development and use of a clinical decision support system for the diagnosis of social anxiety disorder. *Comput. Methods Programs Biomed.* **2020**, *190*, 105354. [[CrossRef](#)]
128. Huang, W. Segmentation and diagnosis of papillary thyroid carcinomas based on generalized clustering algorithm in ultrasound elastography. *J. Med. Syst.* **2020**, *44*, 1–8. [[CrossRef](#)]
129. Janarthanan, R.; Doss, S.; Baskar, S. Optimized unsupervised deep learning assisted reconstructed coder in the on-nodule wearable sensor for human activity recognition. *Measurement* **2020**, *164*, 108050. [[CrossRef](#)]
130. Shobha, K.; Savarimuthu, N. Clustering based imputation algorithm using unsupervised neural network for enhancing the quality of healthcare data. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 1771–1781. [[CrossRef](#)]
131. Joloudari, J.H.; Hassannataj Joloudari, E.; Saadatfar, H.; Ghasemigol, M.; Razavi, S.M.; Mosavi, A.; Nabipour, N.; Shamshirb, S.; Nadai, L. Coronary artery disease diagnosis; ranking the significant features using a random trees model. *Int. J. Environ. Res. Public Health* **2020**, *17*, 731. [[CrossRef](#)] [[PubMed](#)]
132. Ardabili, S.F.; Mosavi, A.; Ghamisi, P.; Ferdin, F.; Varkonyi-Koczy, A.R.; Reuter, U.; Rabczuk, T.; Atkinson, P.M. Covid-19 outbreak prediction with machine learning. *Algorithms* **2020**, *13*, 249. [[CrossRef](#)]
133. Pinter, G.; Felde, I.; Mosavi, A.; Ghamisi, P.; Gloaguen, R. COVID-19 pandemic prediction for Hungary; a hybrid machine learning approach. *Mathematics* **2020**, *8*, 890. [[CrossRef](#)]
134. Mahmoudi, M.R.; Baleanu, D.; Band, S.S.; Mosavi, A. Factor analysis approach to classify COVID-19 datasets in several regions. *Results Phys.* **2021**, *25*, 104071. [[CrossRef](#)]
135. Ayoobi, N.; Sharifrazi, D.; Alizadehsani, R.; Shoeibi, A.; Gorriz, J.M.; Moosaei, H.; Khosravi, A.; Nahavi, S.; Chofreh, A.G.; Goni, F.A.; et al. Time Series Forecasting of New Cases and New Deaths Rate for COVID-19 using Deep Learning Methods. *arXiv* **2021**, arXiv:2104.15007.
136. Mahmoudi, M.R.; Heydari, M.H.; Qasem, S.N.; Mosavi, A.; Band, S.S. Principal component analysis to study the relations between the spread rates of COVID-19 in high risks countries. *Alex. Eng. J.* **2021**, *60*, 457–464. [[CrossRef](#)]
137. Mahmoudi, M.R.; Baleanu, D.; Qasem, S.N.; Mosavi, A.; Band, S.S. Fuzzy clustering to classify several time series models with fractional Brownian motion errors. *Alex. Eng. J.* **2021**, *60*, 1137–1145. [[CrossRef](#)]
138. Ardabili, S.; Mosavi, A.; Band, S.S.; Varkonyi-Koczy, A.R. Coronavirus disease (COVID-19) global prediction using hybrid artificial intelligence method of ANN trained with Grey Wolf optimizer. In Proceedings of the 2020 IEEE 3rd International Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE), Budapest, Hungary, 18–19 November 2020; pp. 251–254. [[CrossRef](#)]
139. Kumar, R.L.; Khan, F.; Din, S.; Band, S.S.; Mosavi, A.; Ibeke, E. Recurrent Neural Network and Reinforcement Learning Model for COVID-19 Prediction. *Front. Public Health* **2021**, *9*, 744100. [[CrossRef](#)]
140. Yang, F.; Moayedi, H.; Mosavi, A. Predicting the Degree of Dissolved Oxygen Using Three Types of Multi-Layer Perceptron-Based Artificial Neural Networks. *Sustainability* **2021**, *13*, 9898. [[CrossRef](#)]
141. Qurat-Ul-Ain, F.A.; Ejaz, M.Y. A comparative analysis on diagnosis of diabetes mellitus using different approaches—A survey. *Inform. Med. Unlocked* **2020**, 100482. [[CrossRef](#)]
142. Nosratabadi, S.; Mosavi, A.; Duan, P.; Ghamisi, P.; Filip, F.; Band, S.S.; Reuter, U.; Gama, J.; Gandomi, A.H. Data science in economics: Comprehensive review of advanced machine learning and deep learning methods. *Mathematics* **2020**, *8*, 1799. [[CrossRef](#)]
143. Mosavi, A.; Faghan, Y.; Ghamisi, P.; Duan, P.; Ardabili, S.F.; Salwana, E.; Band, S.S. Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics* **2020**, *8*, 1640. [[CrossRef](#)]
144. Chen, H.; Heidari, A.A.; Chen, H.; Wang, M.; Pan, Z.; Gandomi, A.H. Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. *Future Gener. Comput. Syst.* **2020**, *111*, 175–198. [[CrossRef](#)]
145. Hu, J.; Chen, H.; Heidari, A.A.; Wang, M.; Zhang, X.; Chen, Y.; Pan, Z. Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection. *Knowl.-Based Syst.* **2021**, *213*, 106684. [[CrossRef](#)]
146. Zhang, Y.; Liu, R.; Heidari, A.A.; Wang, X.; Chen, Y.; Wang, M.; Chen, H. Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis. *Neurocomputing* **2021**, *430*, 185–212. [[CrossRef](#)]
147. Zhao, D.; Liu, L.; Yu, F.; Heidari, A.A.; Wang, M.; Liang, G.; Muhammad, K.; Chen, H. Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. *Knowl.-Based Syst.* **2021**, *216*, 106510. [[CrossRef](#)]

148. Tu, J.; Chen, H.; Liu, J.; Heidari, A.A.; Zhang, X.; Wang, M.; Ruby, R.; Pham, Q.V. Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance. *Knowl.-Based Syst.* **2021**, *212*, 106642. [[CrossRef](#)]
149. Dehghani, E.; Ranjbar, S.H.; Atashafrooz, M.; Negarestani, H.; Mosavi, A.; Kovacs, L. Introducing Copula as a Novel Statistical Method in Psychological Analysis. *Int. J. Environ. Res. Public Health* **2021**, *18*, 7972. [[CrossRef](#)] [[PubMed](#)]
150. Shan, W.; Qiao, Z.; Heidari, A.A.; Chen, H.; Turabieh, H.; Teng, Y. Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis. *Knowl.-Based Syst.* **2021**, *214*, 106728. [[CrossRef](#)]