IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 September 2022, accepted 25 September 2022, date of publication 27 September 2022,
date of current version 12 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210347

RESEARCH ARTICLE

Machine Learning Model for Hepatitis C Diagnosis Customized to Each Patient

LERAN CHEN[1,2], PING JI[1], AND YONGSHENG MA[2]
[1]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China
[2]Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen 518055, China

Corresponding authors: Ping Ji (p.ji@polyu.edu.hk) and Yongsheng Ma (mays@sustech.edu.cn)

This work was supported by the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, under Grant RK3L.

ABSTRACT Machine learning is now widely used in various fields, and it has made a big splash in the field of disease diagnosis. But traditional machine learning models are general-purpose, that is, one model is used to evaluate the health status of different patients. A general-purpose machine learning algorithm depends on a large amount of data and requires abundant computing power support, relies on the average level to describe the model performance, and cannot achieve optimal results on a specific problem. In this paper, we propose to train a unique model for each patient to improve the accuracy and ease of use of the model. The proposed approach to solving a problem in the paper is from three perspectives (1) targeted data processing, (2) model structure design: Passing in patient-related information into the model, and (3) hyperparameter tailored optimization. The preliminary experimental results show that using the custom model has advantages of high accuracy, high confidence, and low resource required to diagnose a patient. In the Hepatitis C dataset, over 99% accuracy and 94% recall were achieved using a smaller dataset (only 615 individuals' data) without knowledge of the relevant field. Traditional algorithms such as XGBoost or multi-algorithm ensemble could achieve less than 95% accuracy and only less than 70% recall. Out of a total of 56 patients, the custom model was able to identify 53 patients 20 more than traditional methods, bringing a new and efficient tool for future hepatitis C prevention and treatment efforts.

INDEX TERMS Machine learning, custom model, hepatitis C, disease diagnosis, data augmentation, parameter optimization.

I. INTRODUCTION

Hepatitis C is an undetectable silent killer, a serious disease that is slowly progressive and potentially carcinogenic, and can remain latent in the body for 10-20 years [1], [2]. Typically, only about% of patients with hepatitis C virus infection can recover spontaneously within six months, and 70% of patients turn into chronic viral infection [3]. The hepatitis C virus is extremely stealthy, and WHO estimates that only about one in five of the more than 50 million people living with hepatitis C worldwide are aware that they have the disease, with an underdiagnosis rate of up to 80% [4]. In the early to mid-stages of hepatitis C infection, there are usually no obvious signs and symptoms. Patients may experience dizziness and weakness and poor sleep, which can easily be confused with fatigue caused by work or study [5]. As a result, many patients are often found to have hepatitis C when they are examined for other diseases, and some patients are even found to have hepatitis C when cirrhosis or liver cancer is detected. It is because of this stealthy nature that the damage caused by hepatitis C is chronic and progressive. The hepatitis C virus replicates primarily in the liver cells and damages them [6]. Over time, liver cells in the body will continue to develop inflammation, degeneration and necrosis. There is no vaccine to prevent hepatitis C, so people at risk can only be diagnosed and treated for hepatitis C in a timely manner by taking the initiative to get tested for the hepatitis C virus at the hospital [7], [8]. Although hepatitis C is dangerous, only 1 ml of blood is needed to test for infection with the virus. Once diagnosed, there is no need to panic, as more than 95% of patients with hepatitis C can be cured with standardized and systematic treatment [9], [10] [11], [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

However, the greatest difficulty in the prevention and treatment of hepatitis C disease is that most patients do not know that they have hepatitis C. The mainstream diagnostic tools for hepatitis C are: 1. Liver function tests(LFTs), which assess liver disease from liver-related metabolites [13], [14] [15]. 2. Hepatitis C antibody tests, which clarify whether the body is infected with the hepatitis C virus. If the test result is positive, it indicates that the patient is currently infected with hepatitis C or has previously been infected with hepatitis C [16], [17]. 3. Hepatitis C virus RNA test, this test can effectively determine how long the infection has been present and also how much of the virus is present in the patient's body [18], [19]. 4. Liver puncture or ultrasound: this is the main way to determine the severity of the liver disease. Generally speaking, if the disease is serious or has a long duration, these two tests should be used to analyze the progress of the liver disease, which is also the key to the current treatment process for patients who are diagnosed [20], [21].

Among these tests mentioned above, only liver function tests are easy to perform at regular checkups and have high marginal utility (LFTs can be used to analyze many diseases related to the liver). Antibody and RNA tests are more targeted and less prevalent in general health care facilities, and are relatively costly and not conducive to mass adoption. Puncture tests or ultrasound are generally used to detect the progression of disease in patients with confirmed disease and are not suitable for making early disease diagnosis. This is why making good use of the data from liver function tests has become an effective means of identifying hepatitis C patients earlier.

In order to make the best use of the collected data, a powerful tool such as machine learning is natural. However, the current direction of machine learning is deep learning, which relies on a large number of datasets, which contradicts the small amount of medical-related data accumulated today. Borisov *et al.* points out that deep learning methods have a major disadvantage in the processing of structured data [22], and the performance of deep learning models with huge numbers of participants is even far behind some commonly used tree models [23]. And there are also obvious ethical issues with today's machine learning models when dealing with medical-related problems, as they are judged by their average performance on the validation set. Perhaps the model can perform well on average, but who wants to be the ''unlucky patient'' who is misjudged by the model? Every data sample that is processed by the model is closely related to a patient. The disease diagnostic model is not just discussing the categorization of this data sample, but will actually affect the future of a flesh-and-blood real individual. In order to overcome the above ethical issues, the primary pursuit of a custom model for the selected target patient is the highest possible degree of accuracy. Aim not only for the overall average performance of the model, but also to ensure that the worst performance of each case is acceptable.

In this paper, we propose a machine learning model for hepatitis C diagnosis customized for each patient. The major difference from the traditional model is that the data of the patient to be diagnosed is incorporated into the training process during the training session. The comparison of the customized solution and traditional machine learning is shown in Figure 1 With the help of richer information, the model achieves better accuracy and can correctly categorize almost all patients. The second section analyzes some of the relevant research developments, and the third section describes the dataset used and some basic data processing tools. The fourth section will clarify the principles of the model construction and detail the process as much as possible in order to facilitate the replication of the results by subsequent scholars or medical practitioners. The fifth section presents some experimental results, and the last section will provide some summary and outlook.

II. RELATED WORK

Although this article is a study of the diagnostic issues of hepatitis C, it is essentially an analysis based on medical data already collected and does not involve relevant medical-related knowledge. Therefore, in this section, we will not analyze the virology of hepatitis C and disease-related knowledge, but mainly summarize the existing data analysis tools for the disease and their effects. In this section, we will discuss: 1. the development of structured data processing in the field of machine learning; 2. the customization and lightweighting of traditional models for specific application scenarios to make them easier to use; and 3. the progress of studies using the same dataset.

A. METHODS OF PROCESSING STRUCTURED DATA

1) GRADIENT BOOSTED DECISION TREE

The field of structured data (i.e., tabular data) has historically been dominated by conventional machine learning algorithms like Gradient Boosted Decision Tree (GBDT) [24]due to their better performance [23]. Scientists and businesses alike rely heavily on several GBDT algorithms, the most popular of which being XGBoost, LightGBM [25], and CatBoost [26]. A scalable gradient boosting tree technique, GBDT produces state-of-the-art results on numerous tabular datasets, and XGBoost is one of the most prominent implementations of GBDT. The process known as ''gradient boosting'' builds new models using the residuals of older models to produce more accurate predictions [27], [28]. XGBoost's foundation is the same as GBDT's, but it's been improved upon.For example, the second-order derivative makes the loss function more accurate; the use of regular terms to avoid tree overfitting; Block storage allows parallel computation, etc.

2) DEEP NEURAL MODELS

Since deep neural networks have been so successful in image recognition, numerous recent research have extended deep learning to the area of tabular data, with the goal of improving the performance of tabular data by introducing novel neural architectures [22], [29] [30]. Based on the deep learning ideas

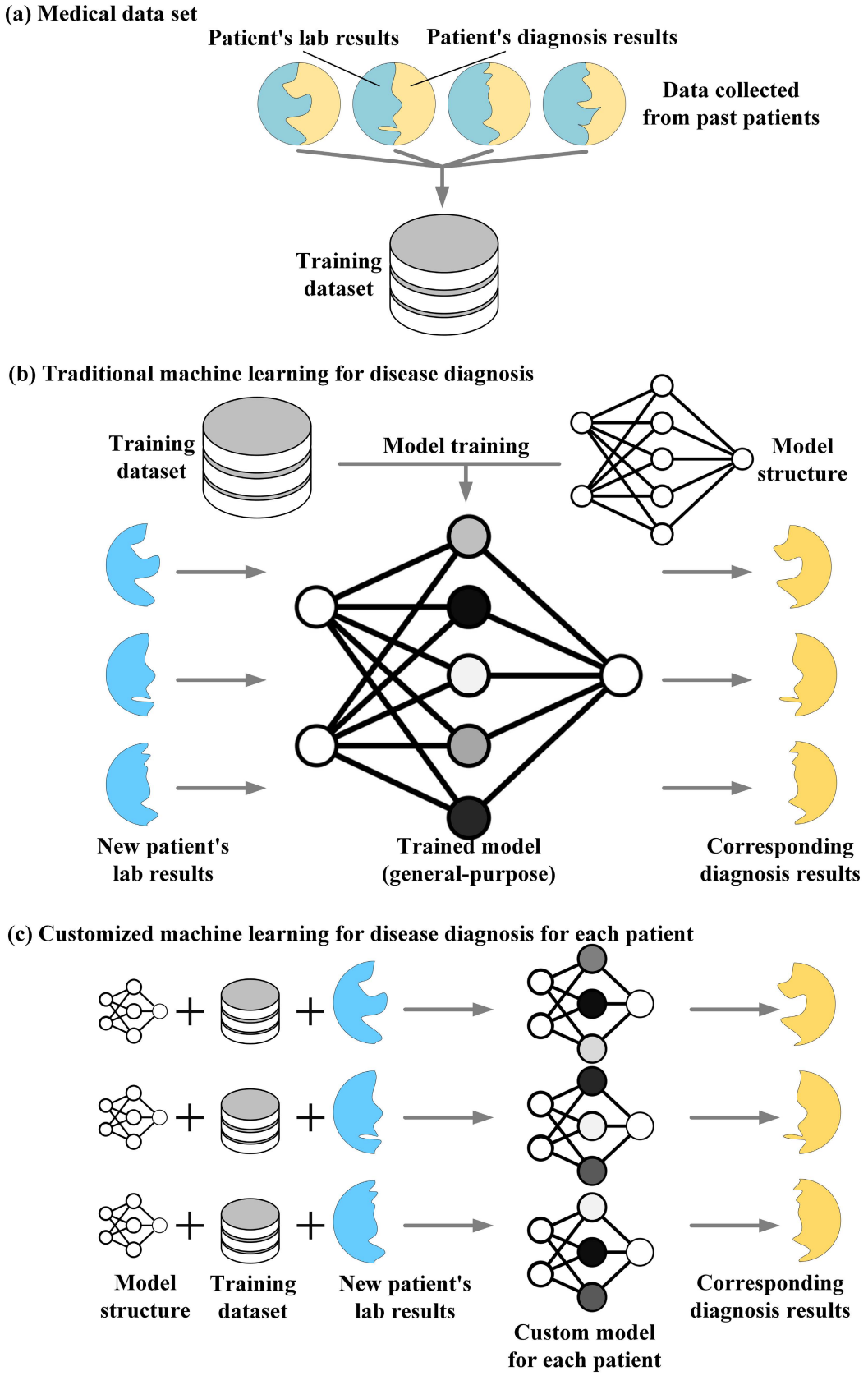

FIGURE 1. Comparison of the customized solution and traditional machine learning.

these models draw from, the models can be classified into two categories.

Attention-based models. Given the novel route taken by attention-based models in deep learning, several researchers have experimented with attention-like modules in tabular deep networks. Two types of focus have recently been proposed: inter-sample attention, where characteristics within a single sample interact, and intra-sample attention, where individual data points make advantage of row-level or sample-level interactions. [31], [32].

Differentiable trees. The series of work presented here seeks to make decision trees differentiable because of the impressive results obtained by decision tree ensembles when applied to tabular data. Due to their lack of differentiability and gradient optimization, classical decision trees are limited in their use in some specific application scenarios. Fortunately, recent research has found a solution to this issue: by making tree functions and tree routing differentiable by smoothing the decision functions in the internal tree nodes differentiable [33], [34].

But even with the improvement of these new approaches and the combination of them, it is still difficult for deep neural models to outperform traditional GBDT across the board in structured data.

B. CUSTOMIZATION AND LIGHTWEIGHTING OF COMPLEX MODELS

With the rapid accumulation of data [35], [36], a variety of all-encompassing datasets have been built [37], [38] [39], the differences between data and compatibility issues were ignored. This neglect leads to the difficulty for complex models for complete scenarios to perform consistently on all problems [40], [41], there will always be particular problems that are substantially off in prediction, and there will always be images that cannot be correctly classified. This leads to the fact that if one wants to apply large proven models to specific particular datasets, that is, to adapt the original models to specific problems, this is not easy to achieve. In the field of problem-based machine learning research, there has been a minimal exploration of this.

Some researchers [42], [43] [44], [45] [46] discusses how existing complex models can be tailored to specific problems, making the original model better applicable to specific datasets using transfer learning. Since traditional mature neural networks are large and bloated. Some researchers [47], [48] [49], [50] [51] attempts to compress the parameters of the model based on the existing model employing knowledge distillation and model simplification to achieve the effect of improving the speed of computing.

There also are several scholars who proposed some tricks for data augmentation [52], [53] [54], [55] [56], which can make the model improve the accuracy of analysis in specific scenario.

However, these solution ideas are still rarely discussed for very specific individual problems, and this paper will try to fill the gap and demonstrate the feasibility.

C. HEPATITIS C DISEASE DIAGNOSIS USING THE SAME DATASET

In the field of medical diagnostics, machine learning has been showing its capabilities since very early on. Back in 2017, Hashem *et al.* compared several ways to predict hepatitis C using blood markers, yielding a best accuracy rate of 66.3% to 84.4% [57]. In 2018, Hoffmann *et al.* collected and organized the dataset used in this paper, several medical researchers analyzed the data through a tree model, yielding an accuracy rate of best 75.3 [58]. This dataset was donated to the UCI Machine Learning Repository in June 2020 [59], [60]. After that, Chicco and Jurman used the dataset to perform Ensemble Learning on the AST/ALT ratio to achieve a 95.4% accuracy rate on whether the disease was present or not [61]. Chawathe *et al.* achieved a 95% accuracy rate and 89% recall rate by fusing multiple models. But for specific applications in medical diagnosis, all this needs to be enhanced [62].

We need to make every effort so that all patients are accurately identified and all healthy patients can be correctly classified without additional biopsies.

III. METHODOLOGY

The algorithm design in this paper is based on thinking from two perspectives, from the perspective of the user of the model and from the perspective of the data.

1) USER's PERSPECTIVE

When a patient's certain laboratory indicator contributes significantly to the outcome of a disease, it means that this indicator is important and should be given attention. This kind of judgment is what experienced physicians are good at, and as someone who has some experience working with key characteristic variables, understanding them is a must. Likewise, for indicators that are not important in the laboratory results, the physician will find that the impact of this indicator is not important in the diagnosis of a particular disease [63], [64] [65].

In general, after a systematic study of medical knowledge and a certain period of internship, a doctor can make a general judgment about various laboratory indicators, which ones are important and which ones do not play a role. However, the superficial cognition of inexperienced doctors is not enough to judge the causal relationship between variables in a short time, so if doctors are allowed to intervene in the screening of data at the early stage of data processing, the inaccuracy of doctors' own judgment will be transferred to the data. Therefore, it is wiser to have physicians with extensive experience review the trained model to check whether the importance differences of the weights in the model are consistent with the objective laws of the real world, so as to ensure the reliability and interpretability of the model. It's also an effective way to get more value out of experienced physicians and allow excellent medical resources to serve more people [66].

2) THE PERSPECTIVE OF DATA

The data itself will naturally present differences in the influence of different variables, and will also show the relationship between different data samples. When it comes to data related to disease diagnosis, the data will then reflect similarities between patients. The general process of machine learning mainly describes the relationship between variables, but not much attention is paid to the relationship between samples. The data processing customized for patients proposed in this study is going to fill this gap and explore how to use the relationship between samples to improve the accuracy of the model. The effect of focusing on some of the key samples can be achieved by modifying the ratio of the number between samples, just like a person focuses on the key information in a scene.

Guided by the above ideas, the algorithm proposed in this paper implements model customization for patients in three stages. 1) data processing stage: targeted sample augmentation. 2) model structure design stage: patient data are skillfully passed to the model. 3) hyperparameter optimization stage: model performance under different hyperparameters are judged by new evaluation criteria. We call an individual patient who needs a disease diagnosis a ''target patient''. Each patient's laboratory results can be considered a sample, and a medical dataset will have a very large number of samples.

The framework of the algorithm is depicted in Figure 2, which is divided into three major parts, they are data processing, model building, and parameter optimization. The yellow box on the left is the acquisition process of the traditional machine learning model, and the blue box on the right is the acquisition process of the custom machine learning model proposed in this paper.

A. TARGETED DATA AUGMENTATION

This paper proposes to adjust the proportion of training samples (targeted data augmentation). The operation of this part is shown in Figure 3.

EDA (Exploratory Data Analysis) [67], [68] is an essential part of machine learning and is the first step that starts after acquiring data. In this process, the original data is explored with as few a priori assumptions as possible, summarizing the structure of the data and presenting specific patterns. For a single feature, the data engineer always expects that the variables under that feature can be uniformly distributed or normally distributed within the data. For the whole sample space, the data engineer always expects that each data point can be uniformly distributed in the sample space (it means that the probability (density) corresponding to each sample point in the whole sample space is equal) [69]. This is because imbalanced data can seriously affect the model's effectiveness and even affect the judgment of the model, good or bad. The accuracy of the model is very high for the high proportion categories, and the deviation of the prediction is exceptionally high for the low proportion categories. Nevertheless, The researcher naively thought to get a good model because the higher proportion categories had a more significant effect on the loss and metric [70].

However, in a dataset containing a large number of samples, there always is only limited sample data in the region that should be focused on. If a laboratory result for patients who need to be diagnosed is introduced in the original sample space, the percentage of samples in the training set that are similar to the target patient is tiny. In order to improve the accuracy of the model for the target patient, the proportion of training samples can be adjusted by targeted data augmentation [71]. In the case of the disease diagnosis problem discussed in this paper, to make the custom model more accurate for the target patient, what is done is to reduce the level of attention to the cases that differ significantly from the target patient and pay extra attention to the cases that very similar to the target patient. This allows the model to be more sensitive in identifying potential patients and also allow the model to make correct judgments when faced with healthy cases [72].

The specific operation is as follows: 1. Find several samples from previously collected case datasets closest to the target patient in the whole sample space; 2. Increase the number of these similar samples by a specific method to occupy a more significant proportion of the entire sample space [73], [74]. After determining the idea of targeted data augmentation, two questions arise: 1. how to describe the similarity between samples in the sample space, that is, how to determine that the laboratory results of two patients are more similar; 2. how to expand the number of similar samples by what means. In machine learning and data mining, the concept of ''statistical distance'' is often introduced to describe the magnitude of differences between individuals and thus evaluate the similarity and class of individuals. Depending on the characteristics of the data, different measures can be used. In general, to define a distance function d(x,y), the following criteria need to be satisfied [75]:

1.Non-negativity:

$$d(x, y) \leq 0 \tag{1}$$

2.Identity of indiscernible:

$$d(x, y) = 0 \Longleftrightarrow x = y \tag{2}$$

3.Symmetry:

$$d(x, y) = d(y, x) \tag{3}$$

4.Triangle inequality:

$$d(x, z) \leq d(x, y) + d(y, z) \tag{4}$$

Based on these criteria, the Euclidean distance [76] was selected, Mahalanobis distance [77], Chebyshev distance [78], Minkowski distance [79], and Bhattacharyya distance [80] as alternative options. After a comparison test, the Mahalanobis distance was finally chosen as the criterion to describe the sample similarity. Its most prominent

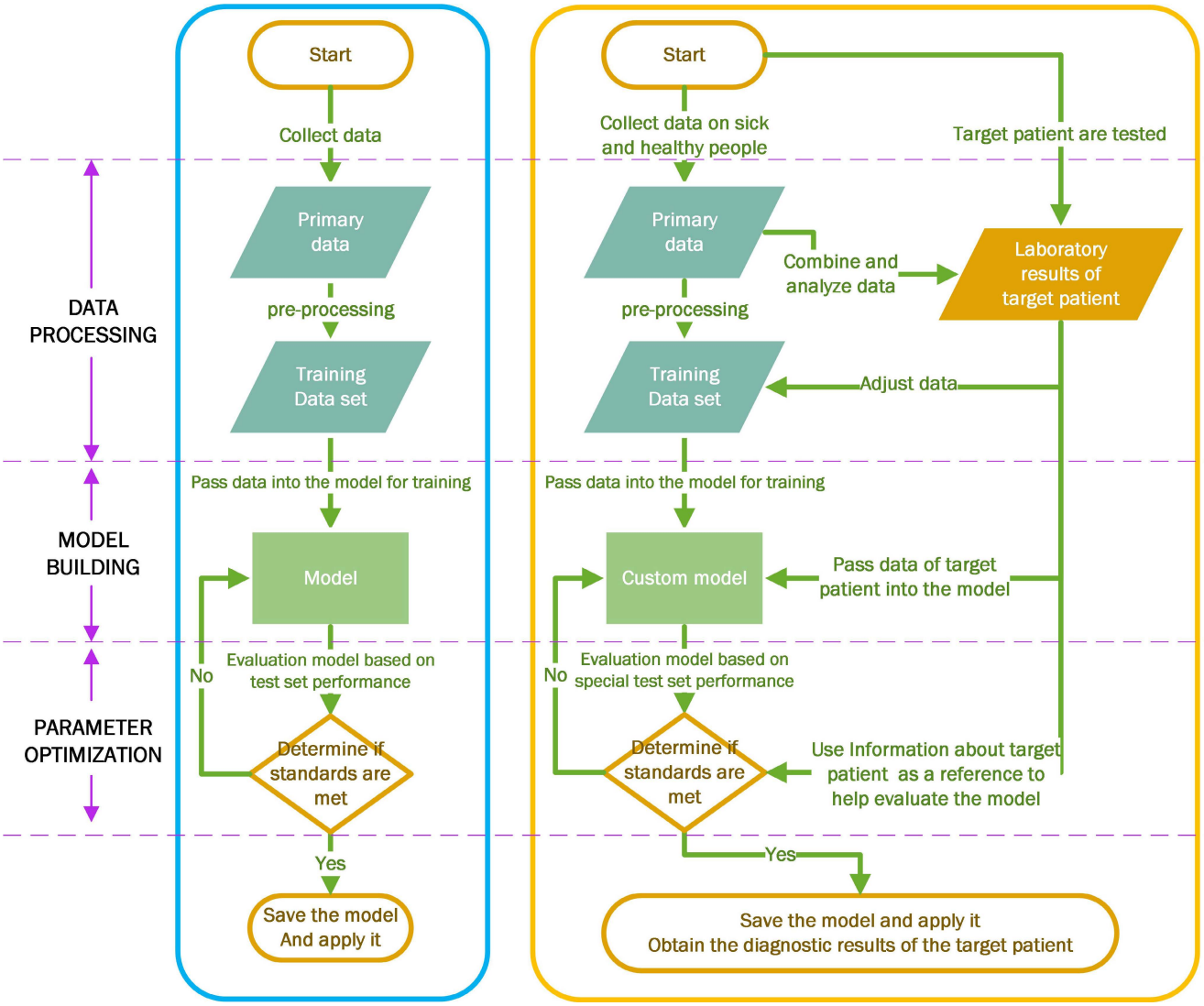

FIGURE 2. Framework of the custom algorithm. The yellow box is the acquisition process of the traditional machine learning model, and the blue box is the acquisition process of the custom machine learning model.

advantage is modifying the traditional Euclidean distance, which corrects the problem of inconsistent and correlated scales of each dimension in the Euclidean distance. It can genuinely reflect the similarity relationship between samples without the constraints of dimensional scales. Other distance criteria in the comparison experiments were more or less influenced from the complex dimensions, resulting in calculated distances that did not satisfy the needs of subsequent experiments. In the future, it will also try to update the similarity criteria in the form of Metric-learning after introducing additional information from professionals. This option allows experienced physicians to judge and score the similarity of patients. An evaluation criterion for evaluating the degree of similarity of patients is then summarized by learning these scores by means of Metric-learning. To increase the number of few samples in the training set that are similar to the target patient to achieve sample balancing, the SMOTE (Synthetic Minority Over-sampling Technique) [81] algorithm was chosen after comparing various methods for adjusting the sample proportions. The SMOTE method is an interpolation-based method that synthesizes new samples for small sample classes. By calculating the Mahalanobis distance between the sample points in the training set and the target patient, a certain stem of samples that are most similar to the target patient is oversampled. The result of this processing is shown in Figure 4. The figure shows a two-dimensional (feature) sample space in which the yellow triangle represents a positive sample and the blue pentagon represents a negative sample. The target patient is to categorize the green squares (target samples) in the sample space. After the Targeted data augmentation process, the samples in the original sample space are targeted augmentation (which can be interpreted as simply copying the samples to increase the weights). The augmentation results in augmenting the samples that are more

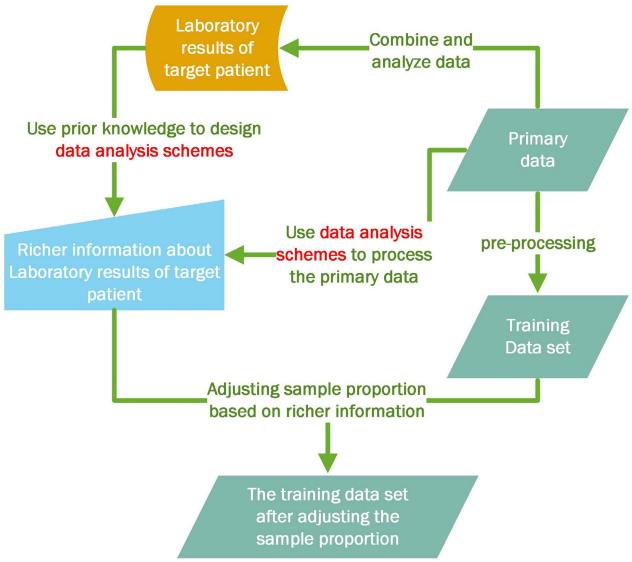

FIGURE 3. Framework of targeted data augmentation.

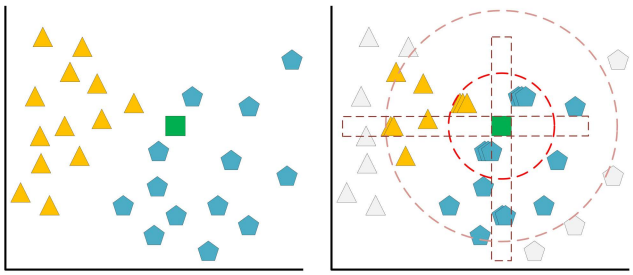

FIGURE 4. Targeted data augmentation effect comparison chart.

similar to the target samples and paying more attention to the samples that are more similar.

Define the data before processing as D shown in equation (5), where there are n samples in total and each sample is differentiated by the i. The features (dimensions) are d in total and are distinguished by the k. The output labels (dimensions) are l in total and are distinguished by the o. So write X and Y in the form of separate matrices as Equation (6).

$$D = (x_k^i, y_o^i) \mid i = 1, 2, \cdots, n; \quad k = 1, 2, \cdots, d; \quad o = 1, 2, \cdots, l \tag{5}$$

$$X \in \mathbb{R}^{n \times d} \quad Y \in \mathbb{R}^{n \times l} \tag{6}$$

After the targeted sample augmentation is performed, it makes the original dataset richer, and here m is defined as the increased number of samples. The new dataset is defined as D_{new} shown in Equation (7). X and Y have also changed as Equation (8).

$$D = (x_k^i, y_o^i) \mid i = 1, 2, \cdots, n+m; \quad k = 1, 2, \cdots, d; \quad o = 1, 2, \cdots, l \tag{7}$$

$$X \in \mathbb{R}^{(n+m) \times d} \quad Y \in \mathbb{R}^{(n+m) \times l} \tag{8}$$

B. CUSTOM MODEL STRUCTURES FOR PATIENTS

This study proposes a form of subtly passing information of target patient to the model under the guidance of the above research idea. The operation of this part is shown in Figure 5.

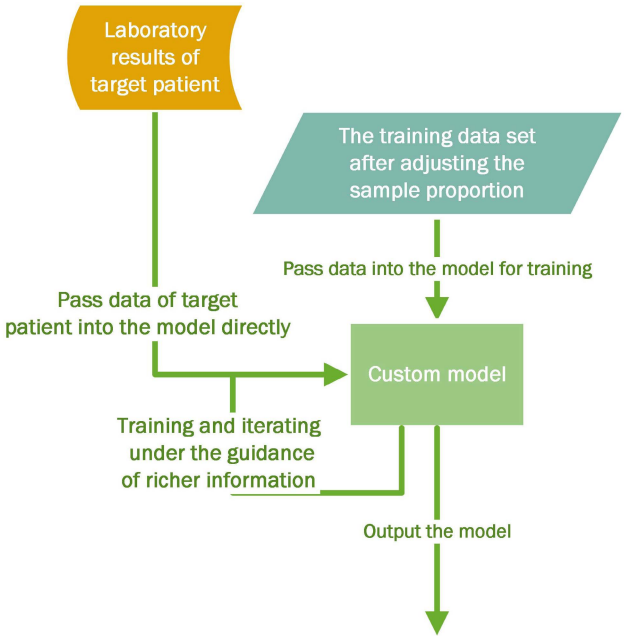

FIGURE 5. Framework of subtly passing scenario information to the model.

Is there any part of a neural network model design that allows the model to receive specific information directly? The answer is yes. Most ordinary algorithms are one-to-one correspondence between input and output; one input gets one output. There is no connection between different inputs. The structure of the traditional neural network is relatively simple: input layer-hidden layer-output layer [82].

RNN [83] is different from the traditional neural network in that each time, the output of the previous time is brought to the next hidden layer and trained together. Inspired by RNN, this study proposes to take the selected target patient as a particular input and bring it into the hidden layer for operation. The biggest advantage of this approach is that it makes the model more sensitive to the target patient right through the training process. And since only one layer of neural network is added, only one hyper parameter that can be pre-set and one parameter that can be trained, there is little impact on the overall complexity of the model.

The specific way is divided into two steps:

After normalizing the data uniformly, the selected target patient is multiplied by a "bias coefficient: e" and added to all the input data of the training set. The "bias coefficient" can be freely set and represents the initial offset to the target patient on the entire training set. e can be positive or negative, with larger absolute values indicating a greater influence on the training set according to the target patient. The physical meaning of this operation in the sample space can be understood as a shift of all sample points in the sample

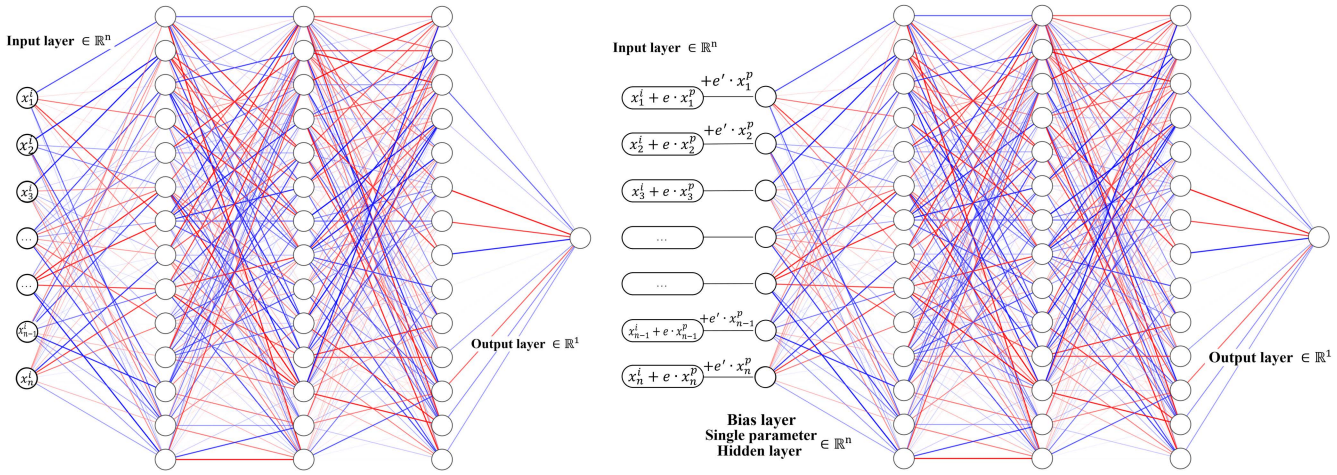

FIGURE 6. Structural comparison between the traditional model and custom model. $\vec{x}^i$ is the ith sample in the training set with total n dimensions. $\vec{x}^p$ is the target patient sample. x_k^p is a scalar in the kth direction of the vector $\vec{x}^p$, and it represents a value for a feature of the target patient.

space in the direction of the selected target patient. If the value of bias coefficient $e = -1$, the origin of the whole sample space coordinate system becomes the selected target patient sample points. A bias layer with only one parameter is added immediately after the input layer. In the bias layer, the input data are multiplied with the selected target patient by a "restore coefficient: e'" and added again. e' can be automatically adjusted during the model training by back-propagation. That is, the only parameter added to the model that can be automatically adjusted during the training process. The Structural comparison between the traditional model and custom model is shown in Figure 6 [84]. Only one layer is added to the model structure, and only one parameter is added that needs to be trained.

In short, the input data is moved twice bias according to the direction of the selected target patient. The first move is a move of the overall training set according to the predefined parameter e. The second move is a move of the samples in the bench during the training process and the training parameter e' by back-propagation at the same time. During this e' iteration, the origin of the coordinate system of the source dataset is displaced back and forth in the direction of the selected target patient, which forces the model to be stable for all sample points in the direction of the selected target patient.

At the beginning of this approach design, it is expected that the restore coefficient e' would gradually converge to the opposite of bias coefficient e during the training process, that is $e' = -e$. When $e' = -e$ is achieved, it means that the input passed into the subsequent hidden layer is original data, and all artificially added bias is counteracted.

In the sample space, in addition to the coordinates of the absolute position which contains all the information about the sample, the direction of the sample is also crucial information. In the process of Adaptive bias adjustment, the directions of almost all samples changes with each change

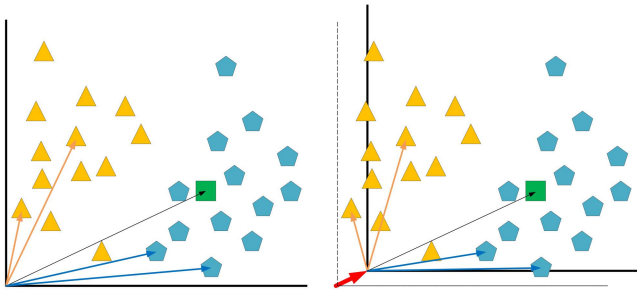

FIGURE 7. Adaptive bias adjustment effect comparison chart.

of e', and only the direction of the target sample is always constant. The change of direction vector of each sample in Figure 7 illustrates this change very visually. The left panel represents the unbiased sample space, while the right panel shows the biased sample space. A comparison of the two plots shows that only the direction of the target patient represented by the green square is stable, while the direction of all other samples has changed.

However, during the experiments, it was found that the final result of e' was mostly negative regardless of whether e was set to positive or negative values by reading the final parameter value e' after iteration. In other words, the model tends to orient the overall sample space to the negative half-axis during the learning process. This situation was analyzed: because the Rectified Linear Unit(ReLU) [85] is used as the activation function used in the subsequent hidden layer, more negative semi-axis variable values will be processed to zero, and the sample space will be more concentrated, making the overall function more likely to converge. In order not to lose the accuracy and separability of the data in the original sample space, a parameter selection procedure for the e value is subsequently introduced. The above method can be easily applied to MLP (Multilayer Perceptron). A simplified MLP

with only one hidden layer is defined, and its operational logic is summarized in the mathematical formula $f(x)$ shown as equation (9), where A_o and A_h represent the activation function of the activation function of the output layer and the hidden layer, respectively. Typically, SoftMax activation functions are used for classification problems and Identity functions are used for regression problems. W_o and W_h represent the weights (also called connection coefficients) of the output layer and the hidden layer, respectively, and b_o and b_h are the biases of the output layer and the hidden layer. The types of A_o and A_h are selected during the model construction phase and can be optimized later as hyperparameters. W_o, W_h, b_o and b_h are iteratively updated during the training process.

$$f(x) = A_o\,(b_o + W_o\,(A_h\,(b_h + W_h\,X))) \\ where\ \ W_h \in \mathbb{R}^{d\times q}\ \ b_h \in \mathbb{R}^{1\times q} \tag{9}$$

After introducing the adaptive bias adjustment into the MLP model, the equation changes as shown in Equation (10). $\vec{x}^p$ is the laboratory report data of target patient to be analyzed, and X_p is obtained by copying $\vec{x}^p$ to the same size as X. Among the two newly added variables, e is selected during the model construction phase and can be optimized later as a hyperparameter. e' is updated iteratively during the training process. Only one scalar, e', that needs to be iterated is added during the training process, and the impact on the number of parameters of the model can be negligible.

$$f(x) = A_o(b_o + W_o(A_h(b_h + W_h(X + eX_p + e'\,X_p\,)))) \\ where\ \ W_h \in \mathbb{R}^{d\times q}\ \ b_h \in \mathbb{R}^{1\times q}\ \ e \in \mathbb{R}^{1\times 1}\ \ e' \in \mathbb{R}^{1\times 1} \tag{10}$$

Adaptive bias adjustment is not directly applicable to multiple patients for the time being. As an alternative, adaptive bias adjustment can be trained for each patient first, and finally, the effect of customization for the selected target patients can be achieved by model fusion. That is, it is possible to customize both to individual patients and to several patients who share common characteristics. For example, mass testing for hepatitis C infection in some alcoholic populations.

C. VALIDATION AND PARAMETER TUNING

Once a model has been built for a single problem, a question arises. How can the effectiveness of this new model be evaluated? In the past, all-purpose models were evaluated by reserving a separate portion of the collected data as the validation set and then tuning the model by evaluating the performance of the trained model on the validation set [86]. However, such a process is no longer applicable to small sample sizes. First of all, samples in a small sample space are already very rare, and each unique sample contributes significantly to the complexity of the entire sample space. Once some samples are stored separately as validation sets and do not participate in the training process, the training effect of the model itself will be greatly affected. If evaluated by K-Fold Cross-Validation, it again suffers from the loss of accuracy when the final model is fitted [87].

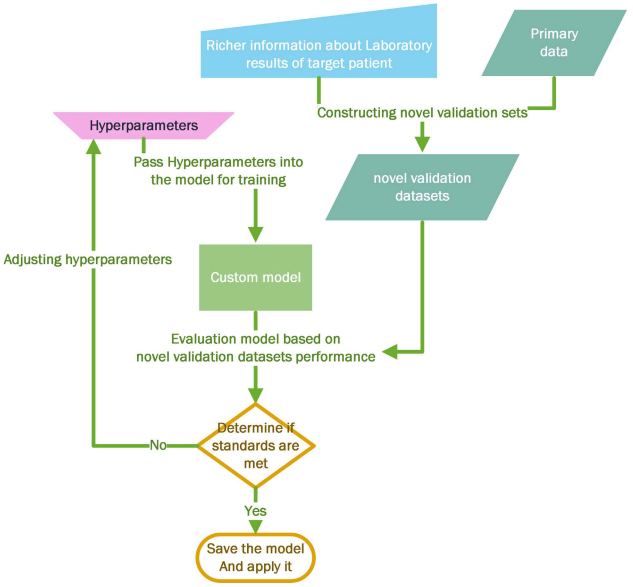

FIGURE 8. The framework of Novel constructs of Validation sets.

Therefore, Novel constructs of Validation sets are proposed in this study. The operation of this part is shown in Figure 8.

When it is necessary to evaluate the excellence of a completed training model, two main criteria are generally used as a reference. One is the loss such as root-mean-square error (RMSE) [88] of the training set and the other is the loss of the validation set. This general case requires us to be able to calculate the loss or RMSE of the validation set, meaning that the correct output of the validation set need to be known. This is possible in the general research and development phase because these validation sets are divided from the complete dataset. But how to evaluate the accuracy of the model for the validation set when nobody has the correct output results of the validation set in the real scenario of the application? This study propose to find a number of samples from the training set that are closest to the selected target patient as the validation set to evaluate the accuracy and stability of the model for the selected target patient [89], [90].

The result of such an operation mainly affects the operation of the loss function [91], the original loss function as in Equation (11).

$$J(W_o, W_h, \gamma, \theta, b_o, b_h) = \frac{1}{2}\sum_{i=1}^{n}\sum_{o=1}^{l}(y_o^{i\prime} - y_o^i)^2 \tag{11}$$

After replacing the new validation set, only the selection of y-values for the loss function formula is changed (as in Equation 12), without adding additional computational effort. The main advantage of this is that it allows the validation set to represent the accuracy and stability of the model for the target patient, rather than the traditional validation set for the entire sample space.

$$J(W_o, W_h, \gamma, \theta, b_o, b_h) = \frac{1}{2}\sum_{i=1}^{n}\sum_{o=1}^{l}(y_{onew}^{i\prime} - y_{onew}^i)^2 \tag{12}$$

TABLE 1. Statistics for each feature of the dataset. STE is the abbreviation for standard error and STD is the abbreviation for standard deviation [59].

feature	mean	median	STE	STD	range
Age	47.41	47	0.405	10.06	[19,77]
Sex	0.55	1	0.020	0.50	[0,1]
ALB	41.55	41.9	0.243	6.01	[0,82.2]
ALP	66.29	65.3	1.134	28.11	[0,416.6]
ALT	28.40	23	1.027	25.47	[0,325.3]
AST	34.79	25.9	1.334	33.09	[10.6,324]
BIL	11.40	7.3	0.793	19.67	[0.8,254]
CHE	8.20	8.26	0.089	2.21	[1.42,16.41]
CHOL	5.28	5.29	0.053	1.31	[0,9.67]
CREA	81.29	77	2.006	49.76	[8,1079.1]
GGT	39.53	23.3	2.204	54.66	[4.5,650.9]
PROT	71.93	72.2	0.247	6.13	[0,90]

1) OPTIMAL PARAMETER SELECTION

With parameters that evaluate the accuracy and stability of the model with respect to the selected target patient as a guide, Hyperparameter optimization of custom models can be carried out with the help of optuna [92], [93] framework. In addition to the usual hyperparameters, such as the number of nodes per layer, epochs, and drop-out ratio, it is found that hyperparametric optimization of the bias coefficient e not only preserves the accuracy and separability of the data in the original sample space as much as possible, but also improves the accuracy of the model for the selected target patient.

2) EARLY TERMINATION OF TRAINING

In the training process of conventional models, training is usually terminated by setting epochs or terminated early when the validation set loss is no longer decreasing [94], [95]. Since the scenario developed in this paper has a more explicit the selected target patient, the model can be called to compute the selected target patient after each iteration and terminate the training early when the output is more stable, or the loss of the validation set is no more significantly decreases.

IV. DATASET AND PREPROCESSING

To cope with the shortcomings of traditional detection means, difficult, costly, and time-consuming, this paper tries to diagnose hepatitis C status through the use of blood biomarkers. The Hepatitis dataset from UCI machine learning repository was selected to show the effectiveness of the custom algorithm. This dataset is about blood biomarkers for hepatitis c virus detection. 615 cases of laboratory values of blood donors and hepatitis C patients and demographic values like age. The target attribute for classification is category (blood donors vs. Hepatitis C). And there are 14 attributes.

Ethical Considerations: The data involved in this paper are all data obtained from publicly available sources [59] and have been properly cited according to the data publisher's requirements. Some of the data related to case information of some patients, where information related to identity has been removed or desensitized by the data publisher so as not to reveal the privacy of the patient.

But obviously, accuracy of previous work is not sufficient for medical applications, so more advanced tools are needed to analyze the data. This section will also analyze this data using some of the most popular algorithms in the field of machine learning classification nowadays, in order to compare the advancedness of the proposed approach in this paper.

Exploratory Data Analysis: Perform basic evaluation checks on the data by calling the functions of pandas [96], NumPy [97]. Load the training and test sets and briefly browse the data: head() + .shape(), get familiar with the relevant statistics of the data by .describe(), get familiar with the data types, view the corresponding data column names, and NAN missing information by .info(). View the presence of NAN for each column to determine missing and abnormal data. Have a preliminary perception of the data. Some basic information and analysis of the data are shown in Table 1.

Handling of abnormal data and missing values [98]: Each kind of data has its own actual meaning behind it. When the data value exceeds the normal range or is a meaningless expression, it needs to be adjusted or supplemented in a targeted way. The dataset used here was reviewed by the medical staff, and there were no obvious abnormal values. For patient data with missing values in the dataset, this study chose to remove them.

The processed data samples have the following features from x1 to x12: Age, Sex, ALB, ALP, ALT, AST, BIL, CHE, CHOL, CREA, GGT, and PROT. Each sample has a label y1 with 1 and 0 for disease or absence of disease, respectively.

The feature selection process in filtered and wrapped feature selection approaches is explicitly decoupled from the learning training process, which allows for more accurate correlation analysis. As the name suggests, correlation analysis involves looking at how closely related two variables are by analyzing them together. Correlation analysis can only be carried out if there is some sort of link or probability between the associated elements. Carl Pearson, a well-known statistician,

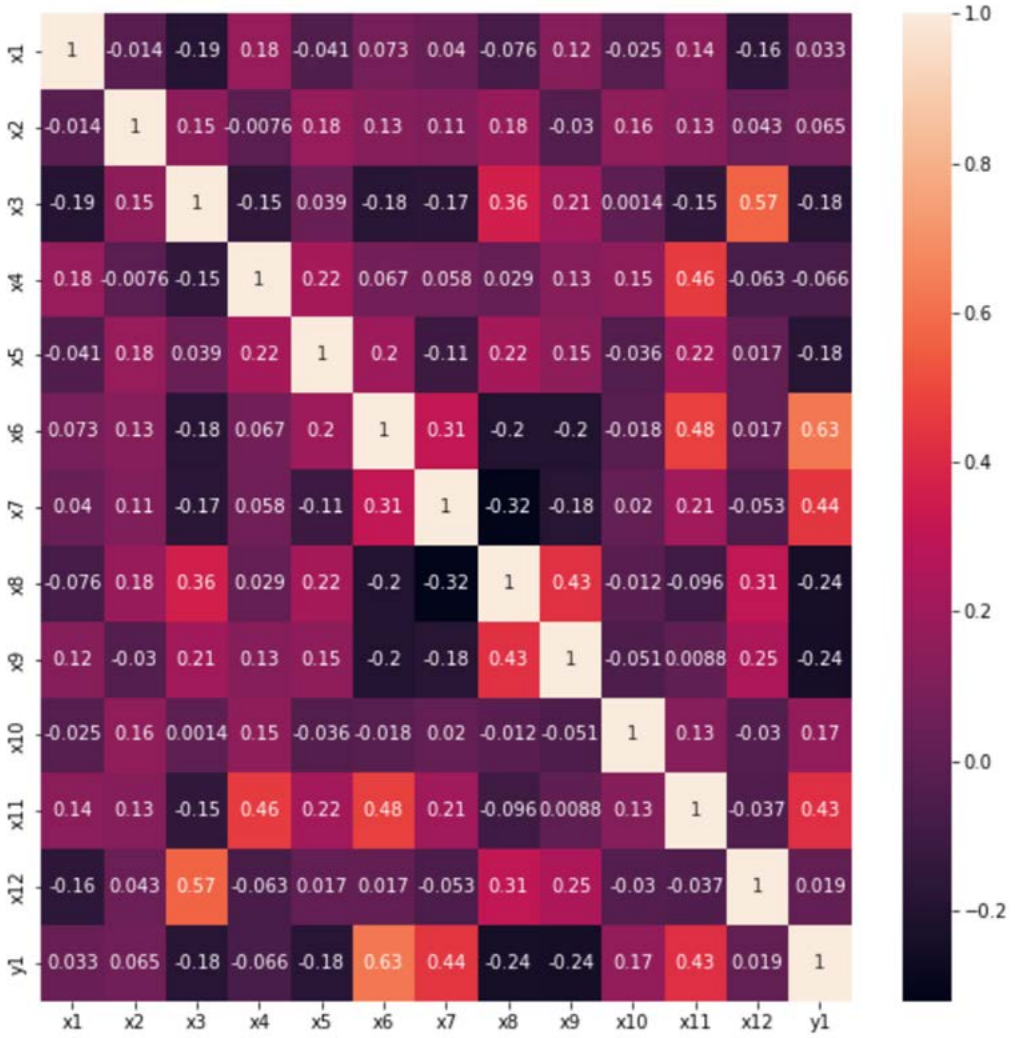

FIGURE 9. Correlation of numeric features.

developed the correlation coefficient [99]. The correlation coefficient is a statistical measure of how strongly two variables are related to one another. By multiplying the two deviations from their respective means, the product-difference approach yields the correlation coefficient; this method is especially useful for calculating the linear single correlation coefficient. Use of the seabon visualization package to create a scatter plot of the correlation analysis matrix, shown in Figure 9 [100].

To be precise, -1 to $+1$ describes the range of the correlation coefficient. In terms of its characteristics, it has the following [101]. Positive correlation between two variables is shown by a r value greater than zero, whereas negative correlation is indicated by a r value less than zero. When $|r| = 1$, there is a perfect linear correlation between the two variables; in other words, they are functionally related. If $r = 0$, then there is no linear relationship between the two metrics. Whenever $0 < |r| < 1$, linear correlation exists between the two variables. As $|r|$ approaches 1 (perfect linearity), the relationship strengthens; as it approaches 0 (poor linearity), the relationship weakens.

Generally, it can be divided into three levels: $|r| < 0.4$ for low linear correlation; $0.4 \leq |r| < 0.7$ for significant correlation; and $0.7 \leq |r| < 1$ for high linear correlation [102]. The correlation analysis revealed that the x6 feature (AST, Aspartate aminotransferase) is very important for the final label. There are also x7 (BIL, Bilirubin), and x11 (GGT, Gamma-Glutamyl Transferase) that contribute to some extent. There is also a clear correlation between features x3 (ALB, Albumin) and x12(PROT, Protein). Visualization of the relationship between digital features based on correlation analysis and several common means of preliminary data analysis were also used to gain a preliminary understanding of the data, but no modifications were made to the data at this stage.

There seems to be a lot of noise/outliers [103]. Some data engineers choose to remove outliers at a fixed rate and

then normalize the data to facilitate analysis [104]. However, considering that this is a medical dataset, all the data is kept in this case to ensure that the data can cover more rare cases. The main purpose of feature engineering is to improve the performance of machine learning by transforming data into features that better represent the underlying problem. Outliers are processed to remove noise and features are constructed to enhance the representation of the data. In order to better enable the use of machine learning models by people who do not have a rich industry background, no additional knowledge is introduced in this case to perform complex processing of the data.

Because of the limited amount of data in the medical dataset, each patient's data information is very precious. Therefore, in order to make full use of this information, the training set is divided in a special way. Each time a specific patient is analyzed, we define the patient's laboratory results as a separate test set and assign all the remaining data to the training set. Whenever a patient changes, the training set changes as well. This is designed to mimic the actual scenario of hospital diagnosis, i.e., for a new patient seeking medical treatment, all the previously saved analysis data is used as the training set to train the model for the new patient.

V. EXPERIMENTS

The experiment will be divided into two phases, the first phase is the comparison experiment phase and the second phase is the hyperparameter tuning experiment phase. The comparison experiment phase is to verify the effectiveness of the custom model and compare the performance of the custom model with other commonly used models on some evaluation criteria. The second phase is to show the extreme performance level of the custom model by tuning for some hyperparameter settings. The experiments were conducted on workstation with an Intel Xeon W-2125 CPU, Quadro RTX 4000 with 8 GiB video memory,32 GiB of DDR4 RAM, and an SSD for secondary storage. All experiments were performed multiple times and the average results were recorded.

A. PARAMETERS OF THE APPROACHES

The Three main improvement approaches are presented in the methodology phase, all of which introduce some new hyperparameters that were not present during the construction of the original machine learning model. Some of these hyperparameters are presented and analyzed next. In the first phase of the experiments, the parameters of the improvement approaches were chosen using invariant parameter settings to verify the generalizability of the improvement scheme. The following is a description of the special parameters.

1) TARGETED DATA AUGMENTATION

a: SIMILARITY THRESHOLD

By calculating the Mahalanobis distance, the degree of similarity between the samples in the training set and the selected target patient can be obtained, and the smaller the value of the Mahalanobis distance, the more similar it is. A threshold value is set in order to facilitate that samples with a Mahalanobis distance less than the threshold value are identified as extremely similar to the selected target patient, and smote oversampling is performed on these extremely similar samples. In this experiment phase, the similarity threshold was set to a fixed value of 2.5.

b: THE PROPORTION OF MINORITY CLASSES AFTER OVERSAMPLING

The samples identified as extremely similar to the selected target patient were oversampled and expanded. The number of expanded minority classes accounted for the majority of samples (samples considered less similar) up to a set value. In this experiment phase, the proportion of the oversampled minority class was set at a fixed 0.3.

2) NOVEL CONSTRUCTS OF VALIDATION SET

The Number of Results Identified as Similar: The samples in the training set are sorted from smallest to largest by calculating the Mahalanobis distance. The number of similar results is set, and the samples that are most similar to the selected target patient are copied from the training set according to the number of similar results to form the validation sets. In this stage, this parameter is set to a fixed number of 10, i.e., the ten samples that are most similar to the selected target patient are selected as the validation set.

3) ADAPTIVE BIAS ADJUSTMENT

Bias Coefficient(e): As introduced in 3.2 above. In this phase, the bias coefficient is set to a fixed -0.2.

B. MODEL CONSTRUCTION

The most basic Back Propagation neural network model (MLP, multilayer perceptron) with three sequential fully connected layers is chosen as the backbone network [105]. Based on the number of independent variables, the number of nodes in each of the three fully connected layers is set to 120, and each layer is output using the activation function ReLU. The final output layer has only one node and use sigmoid as activation function. Total 30,722 trainable parameters. The Adam [106] optimizer, binary_crossentropy, is chosen as the loss function. The overall model construction is simplified as much as possible to evaluate the merit of the final output without using complex techniques. Callbacks are used for the model, val_loss is used as the monitored quantity, and the optimal model is saved. The batch size is chosen to be 256, and the maximum epochs are 100.

C. COMPARISON MODEL SELECTION

After completing the processing of the data, the initial screening of the algorithm was performed with the help of the AutoGluon platform [107]. First, the TabularPredictor and TabularDataset classes of AutoGluon are imported, and then the training data are loaded into the AutoGluon TabularDataset object [108]. Next, AutoGluon is used to

automatically train different models based on different algorithms, and the trained models are used to evaluate model performance by making predictions on the reserved test set data. And XGBoost and LightGBM are the most two efficient algorithm, so the XGBoost and LightGBM model is chosen as a reference.

XGBoost is a very mature algorithm that can be called directly through the XGBoost interface. The dataset is divided into a training set and a test set in the ratio of 0.2(15 patients, 108 healthy people need to be distinguished). The model is set as follows: model_xgboost = XGBClassifier (colsample_bytree=0.7, learning_rate = 0.03, n_estimators=100, subsample=0.7, alpha=0.9) [109], [110]. The results are shown in Table 2.

TABLE 2. Confusion matrix of XGBoost model. Accuracy = 0.9431, Recall = 0.533, Precision = 1, f1-score = 0.6813.

actual \ predict	Hepatitis=1	Hepatitis=0
Hepatitis=1	8	7
Hepatitis=0	0	108

As the confusion matrix demonstrates, the model doesn't really do a good job. It mainly predicted everything as class 0, so Randomized Search was introduced to try to improve this [111], [112] [113]. The settings for Randomized Search are as follows: params = {'learning_rate': [0.01, 0.05, 0.1,0.2], 'max_depth': [3, 4, 5], 'min_child_weight': [1,3,5,7], 'gamma': [0.0, 0.1, 0.2, 0.3], 'colsample_bytree': [0.3, 0.4, 0.5], 'n_estimators': [750, 1000]}. The new XGBoost model is set as follows: model_xgboost = XGBClassifier (); random_search = RandomizedSearchCV (model_xgboost, param_distributions = params, n_iter = 3, cv = 3, scoring = 'accuracy', n_jobs=-1, verbose= 3). The new results are shown in Table 3, there is indeed some progress.

TABLE 3. Confusion matrix of XGBoost model with Randomized Search. Accuracy = 0.9593, Recall = 0.6667, Precision = 1, f1-score = 0.7867.

actual \ predict	Hepatitis=1	Hepatitis=0
Hepatitis=1	10	5
Hepatitis=0	0	108

Similar to the process of XGBoost optimization, the best results obtained by LightGBM after several hyperparameters optimization are shown in Table 4 [114], [115].

TABLE 4. Confusion matrix of LightGBM moedel after hyperparameters optimization. Accuracy = 0.9431, Recall = 0.6667, Precision = 0.8333, f1-score = 0.7811.

actual \ predict	Hepatitis=1	Hepatitis=0
Hepatitis=1	10	5
Hepatitis=0	2	106

D. CUSTOM MODEL PERFORMANCE

1) THE ORIGINAL PERFORMANCE

Next, it is time for the custom model to make its appearance. In the context of this problem, the laboratory result information is mainly unique to each patient. This session focuses on experimenting with combinations of parameters involved in the three improvement approaches so that the most accurate results can be obtained for each selected target patient. The parameter selection phase has three rounds. In the first round, the number of nodes per layer, epochs, and batch size is determined based on the problem complexity, the number of parameters, and the number of samples. In the second round, repeatable experiments are conducted on a certain number of samples to find parameters that are common to the whole dataset: Similarity threshold, Number of results identified as similar, and Proportion of minority classes after oversampling. These parameters are all closely related to the distribution pattern of the samples in the overall sample space. These parameters are determined as fixed values, which basically satisfy all the selected target patients. In the third round, bias coefficient, and drop-out ratio are then selected according to each selected target patient by the optuna framework. Optuna framework is actually a repetitive experiment for multiple parameters, and the optimal parameter is output according to the amount of monitoring. The monitored quantity selected is the MAE of the validation set. The variation interval of bias coefficient is from -0.5 to 0.5, and the variation interval of the drop-out ratio is from 0 to 0.3. The results of the optimal parameters: similarity threshold:3, Number of results identified as similar:5, and Proportion of minority classes after oversampling:0.3. Subsequently added judgment conditions. 1. Stop targeted data augmentation when there are less than 6 samples below the Similarity threshold. 2. When there are more solutions below the Similarity threshold, the data augmentation selects up to 15 samples as the expansion base.

For the same test set samples as the XGBoost and LightGBM model, each sample is treated as the selected target patient, and the custom model is constructed and classified for each the selected target patient in turn, and the final results are as Table 5.

TABLE 5. Confusion matrix of custom model. Accuracy = 0.9919, Recall = 0.9333, Precision = 1, f1-score = 0.9617.

actual \ predict	Hepatitis=1	Hepatitis=0
Hepatitis=1	14	1
Hepatitis=0	0	108

It can be seen that there is a significant improvement over XGBoost and LightGBM, and the entire dataset is iterated in order to better verify the applicability of the method. For the whole dataset, each sample is treated as the selected target patient, and the custom model is constructed and classified for each the selected target patient in turn, and the final results are as Table 6.

TABLE 6. Confusion matrix of custom model on the entire dataset. Accuracy = 0.9864, Recall = 0.875, Precision = 0.98, f1-score = 0.9274.

actual \ predict	Hepatitis=1	Hepatitis=0
Hepatitis=1	49	7
Hepatitis=0	1	531

2) THE PERFORMANCE WITH UPGRADED APPROACHES

By analyzing the results of each selected target patient, the following conclusions can be drawn. Although the results have been good, it can be still found that: in the real-world environment, the unevenness of the sample space can cause much trouble for the custom model. (1) In the case of insufficient similar samples, if the less similar samples are forcibly selected as the benchmark for augmentation, it will increase the density of the overall training set in the region that deviates from the selected target patient. It is more sensible to turn off the target sample augmentation at this time. (2) In the case of too many similar samples, the similar sample set will be more evenly distributed in the overall sample space because of its more significant number. Then, the similar sample set no longer has the sensitivity to the selected target patient. At this time, the overall performance of the validation set cannot accurately reflect the accuracy of the custom model for the selected target patient. (3) For some of the selected target patients with a strange distribution, the most similar samples may be of the opposite category. This strange case can not be distinguished by the custom model for the time being, and more comprehensive and rich balanced data are needed to solve this strange case.

The new results are shown in Table 7 after introducing the automatic disablement of the targeted sample augmentation and the setting of the upper limit of similarity samples.

TABLE 7. Confusion matrix of custom model on the entire dataset with upgraded approaches. Accuracy = 0.9949, Recall = 0.9464, Precision = 1, f1-score = 0.9701.

actual \ predict	Hepatitis=1	Hepatitis=0
Hepatitis=1	53	3
Hepatitis=0	0	532

3) THE PERFORMANCE FOR HIGHER RECALL

Among the target application scenarios of this study, especially when the model is applied to large-scale screening, accuracy is certainly a crucial evaluation criterion. The highest possible accuracy rate allows patients to be accurately identified and treated, and also eliminates the need for additional follow-up testing in healthy individuals.

But the situation changes when hospitals are allowed to diagnose patients who visit them through custom models. For each patient, a false negative poses a much greater risk than a false positive, so it is important to improve the recall rate of the model as much as possible. Guided by such a specific need, the evaluation criteria of the custom model were adjusted. A partial modification of the binary_crossentropy used for the loss function is to make the model consider that the penalty for false negatives is greater than that for false positives. With such an adjustment, the performance of the model in the test set new is shown in Table 8.

TABLE 8. Confusion matrix of custom model for higher recall. Accuracy = 0.9268, Recall = 1, Precision = 0.625, f1-score = 0.9620.

actual \ predict	Hepatitis=1	Hepatitis=0
Hepatitis=1	15	0
Hepatitis=0	9	99

It can be seen that all patients in this test set were correctly identified, but this also led to a more significant decrease in other evaluation criteria. In the recall enhancement experiment on the total data, 55 patients could be identified out of a total of 56 patients.

VI. CONCLUSION

This study combines target patient analysis into a three-stage process of machine learning data processing, model building, and parameter optimization. The first stage of data processing: Targeted data augmentation is performed on the training data considering the patients information, so that the dataset generates relevance according to the target patient. By calculating the relevant parameters such as the Mahalanobis distance, the relevant information within the data is fully explored. And the important weight of the samples closely related to the target patient is increased according to scenario requirement. In this process, the target patient information provides the optimization direction for data processing. The second stage is the training model, which uses the target patient information as an additional fixed training data to achieve the target patient as a constraint at all times during the training process. The goal of this model is to have better performance in specific target patient, which is different from the goal of previous models that emphasize broad adaptability. In this process, the target patient information provides additional information for model training. The third stage of parameter optimization uses the target patient information as a reference standard. This criterion can both verify the magnitude of the error after each iteration and back-propagate the model for tuning based on the magnitude of the error, and compare the advantages and disadvantages between several approaches after all training is completed. It provides a reliable reference for parameter tuning related to the target patient, and it is worthwhile to conduct some interesting and meaningful research on them.

In the testing of the hepatitis C dataset, an extremely accurate model (accuracy of 99.4%) was built without introducing additional information and without having any relevant medical background at all. Comparison of test results among various models is shown in Figure 10. This far exceeds the decision tree model based on expert system logic used in the

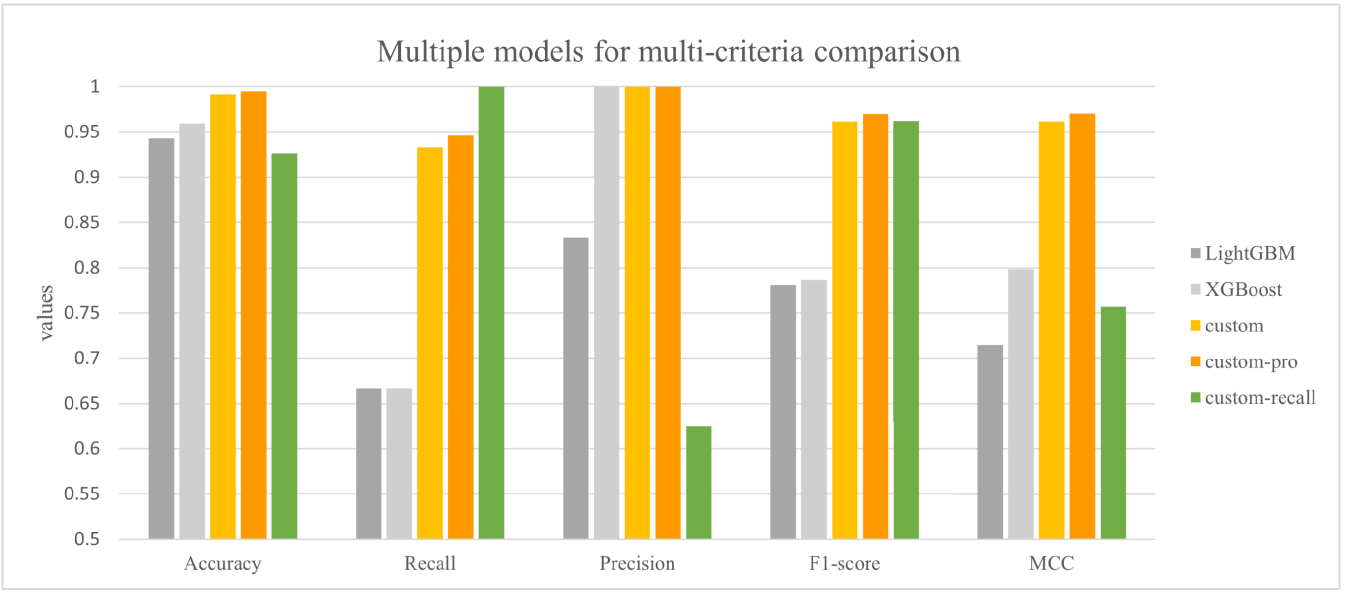

FIGURE 10. Comparison of experiments results among various models. LightGBM and XGBoost models are the result of parameter optimization. For the five models compared, all of them use the same test set, except for custom-pro which is the result obtained on the whole dataset. It can be seen that the custom and custom-pro model show advantages in various metrics, and costom-recall shows that the customized solution can achieve almost 100% recall at the expense of some of the remaining criteria.

TABLE 9. Richer comparison of experimental results. The custom models at the top of the table are the ones proposed in this paper, the models in the middle of the table are the results of tuning and optimization using the mature algorithm, and the models at the bottom of the table are the results of other teams using the same dataset. RF is short for Random Forest, LR is short for Linear Regression, DT is short for Decision Tree, and KNN is short for K-NearesNeighbor.

Model	Accuracy	Recall	Precision	F1-score	Author
Custom-pro	**0.9949**	0.9464	**1**	**0.9701**	Leran et al.
Custom-recall	0.9268	**1**	0.625	0.9620	Leran et al.
CatBoost	0.9593	0.7333	0.9167	0.8313	
XGBoost	0.9593	0.6667	1	0.7867	
RFGini	0.9512	0.6667	0.9091	0.7839	
LightGBM	0.9431	0.6667	0.8333	0.7811	
RF	0.9187	0.6	0.6923	0.7259	
KNN	0.8862	0.6	0.5294	0.7155	
AST/ALT ratio	0.954	0.68	0.993	0.785	Chicco, D. et al. [61]
Fusion model	0.9545	0.8971	0.743	0.9249	Chawathe, S. S. [62]
RF and LR	0.9619	0.7153	0.9694	0.9592	Li, T. H. S. et al. [117]
Ensemble model	0.9559				Edeh, M. O. et al. [118]
Rule-based DT	0.753				Hoffmann, G. et al. [58]

article by the original provider of the data with a medical background. A richer comparison of experimental results is detailed in Table 9.

In testing the hepatitis C dataset, the custom model outperformed the extremely well-developed XGBoost and LightGBM model (selected by AutoGluon). This efficient and accurate model does not require cumbersome tuning and data processing, and does not require medical practitioners to master complex machine learning techniques to use it directly to aid diagnosis.

The analysis time for each individual patient from this part of the case study is about 30s, which can meet the time requirement for medical diagnosis when a patient has finished the blood test. The equipment requirements involved in the training and analysis of the model are very common and easy to implement. This means that there is no need for a separate viral test, and that only the simplest of blood tests are needed to detect the vast majority of patients with hepatitis C, providing a powerful tool for hepatitis C disease control. And this custom model can discard some of the accuracy to

achieve higher recall as needed, and the confidence level is greatly improved and no longer relies on the general average level of the model for evaluation, which ensures that the model can be applied to the treatment of specific patients.

REFERENCES

[1] T. Poynard, M.-F. Yuen, V. Ratzin, and C. L. Lai, "Viral hepatitis C," *Lancet*, vol. 362, no. 9401, pp. 2095–2100, 2003.
[2] M. J. Alter, "Epidemiology of hepatitis C," *Hepatology*, vol. 26, no. S3, p. 62S–65S, 1997.
[3] P. J. Scheuer, P. Ashrafzadeh, S. Sherlock, D. Brown, and G. M. Dusheiko, "The pathology of hepatitis C," *Hepatology*, vol. 15, no. 4, pp. 567–571, 1992.
[4] *Global Hepatitis Report 2017*, World Health Org., Geneva, Switzerland, 2017.
[5] L. B. Seeff, "Natural history of chronic hepatitis C," *Hepatology*, vol. 36, no. 5B, pp. s35–s46, Nov. 2002.
[6] G. M. Lauer and B. D. Walker, "Hepatitis c virus infection," *New England J. Med.*, vol. 345, no. 1, pp. 41–52, 2001.
[7] H. Hagan, E. R. Pouget, and D. C. Des Jarlais, "A systematic review and meta-analysis of interventions to prevent hepatitis c virus infection in people who inject drugs," *J. Infectious Diseases*, vol. 204, no. 1, pp. 74–83, Jul. 2011.
[8] G. J. MacArthur, E. van Velzen, N. Palmateer, J. Kimber, A. Pharris, V. Hope, A. Taylor, K. Roy, E. Aspinall, D. Goldberg, T. Rhodes, D. Hedrich, M. Salminen, M. Hickman, and S. J. Hutchinson, "Interventions to prevent HIV and hepatitis c in people who inject drugs: A review of reviews to assess evidence of effectiveness," *Int. J. Drug Policy*, vol. 25, no. 1, pp. 34–52, Jan. 2014.
[9] M. G. Ghany, D. B. Strader, D. L. Thomas, and L. B. Seeff, "Diagnosis, management, and treatment of hepatitis C: An update," *Hepatology*, vol. 49, no. 4, p. 1335, 2009.
[10] N. J. Burstow, Z. Mohamed, A. I. Gomaa, M. W. Sonderup, N. A. Cook, I. Waked, C. W. Spearman, and S. D. Taylor-Robinson, "Hepatitis C treatment: Where are we now?" *Int. J. Gen. Med.*, vol. 10, p. 39, Oct. 2017.
[11] D. B. Strader, T. Wright, D. L. Thomas, and L. B. Seeff, "Diagnosis, management, and treatment of hepatitis c," *Hepatology*, vol. 39, no. 4, pp. 1147–1171, Mar. 2004.
[12] A. Kohli, A. Shaffer, A. Sherman, and S. Kottilil, "Treatment of hepatitis C: A systematic review," *J. Amer. Med. Assoc.*, vol. 312, no. 6, pp. 631–640, Aug. 2014.
[13] J. K. Limdi, "Evaluation of abnormal liver function tests," *Postgraduate Med. J.*, vol. 79, no. 932, pp. 307–312, Jun. 2003.
[14] S. Gowda, P. B. Desai, V. V. Hull, A. K. Avinash, S. N. Vernekar, and S. S. Kulkarni, "A review on laboratory liver function tests," *Pan Afr. Med. J.*, vol. 3, p. 17, Jun. 2009.
[15] B. R. Thapa and A. Walia, "Liver function tests and their interpretation," *Indian J. Pediatrics*, vol. 74, no. 7, pp. 663–671, Jul. 2007.
[16] E. McGibbon, K. Bornschlegel, and S. Balter, "Half a diagnosis: Gap in confirming infection among hepatitis c antibody-positive patients," *Amer. J. Med.*, vol. 126, no. 8, pp. 718–722, Aug. 2013.
[17] W. Tang, W. Chen, A. Amini, D. Boeras, J. Falconer, H. Kelly, R. Peeling, O. Varsaneux, J. D. Tucker, and P. Easterbrook, "Diagnostic accuracy of tests to detect hepatitis C antibody: A meta-analysis and review of the literature," *BMC Infectious Diseases*, vol. 17, no. S1, pp. 39–57, Nov. 2017.
[18] J.-M. Pawlotsky, M. Bouvier-Alias, C. Hezode, F. Darthuy, J. Remire, and D. Dhumeaux, "Standardization of hepatitis C virus RNA quantification," *Hepatology*, vol. 32, no. 3, pp. 654–659, Sep. 2000.
[19] R. R. Al Olaby and H. M. Azzazy, "Hepatitis c virus RNA assays: Current and emerging technologies and their clinical applications," *Expert Rev. Mol. Diag.*, vol. 11, no. 1, pp. 53–64, Jan. 2011.
[20] E. Caturelli, A. Giacobbe, D. Facciorusso, M. Bisceglia, M. R. Villani, D. A. Siena, S. Fusilli, M. M. Squillante, and A. Andriulli, "Percutaneous biopsy in diffuse liver disease: Increasing diagnostic yield and decreasing complication rate by routine ultrasound assessment of puncture site," *Amer. J. Gastroenterol.*, vol. 91, no. 7, Jul. 1996.
[21] G. Kalambokis, P. Manousou, S. Vibhakorn, L. Marelli, E. Cholongitas, M. Senzolo, D. Patch, and A. K. Burroughs, "Transjugular liver biopsy–indications, adequacy, quality of specimens, and complications—A systematic review," *J. Hepatol.*, vol. 47, no. 2, pp. 284–294, Aug. 2007.
[22] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," 2021, *arXiv:2110.01889*.
[23] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Inf. Fusion*, vol. 81, pp. 84–90, May 2022.
[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3149–3157.
[26] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 6639–6649.
[27] Y. Zhao, G. Chetty, and D. Tran, "Deep learning with XGBoost for real estate appraisal," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 1396–1401.
[28] S. Badirli, X. Liu, Z. Xing, A. Bhowmik, K. Doan, and S. S. Keerthi, "Gradient boosting neural networks: GrowNet," 2020, *arXiv:2002.07971*.
[29] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 972–981.
[30] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular data modeling using contextual embeddings," 2020, *arXiv:2012.06678*.
[31] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein, "SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training," 2021, *arXiv:2106.01342*.
[32] S. O. Arík and T. Pfister, "TabNet: Attentive interpretable tabular learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 6679–6687.
[33] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulo, "Deep neural decision forests," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1467–1475.
[34] S. Popov, S. Morozov, and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–12.
[35] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, pp. 1–16, 2016.
[36] M. Hilbert, "Big data for development: A review of promises and challenges," *Develop. Policy Rev.*, vol. 34, pp. 135–174, Jan. 2016.
[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[39] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, and R. Soricut, "Findings of the 2014 workshop on statistical machine translation," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, pp. 12–58.
[40] G. Harrison, J. Hanson, C. Jacinto, J. Ramirez, and B. Ur, "An empirical study on the perceived fairness of realistic, imperfect machine learning models," in *Proc. Conf. Fairness, Accountability, Transparency*, 2020, pp. 392–402.
[41] C. J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. S. Corrado, and M. C. Stumpe, "Human-centered tools for coping with imperfect algorithms during medical decision-making," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–14.
[42] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. Hershey, PA, USA: IGI Global, 2010, pp. 242–264.
[43] W. Dai, O. Jin, G.-R. Xue, Q. Yang, and Y. Yu, "EigenTransfer: A unified framework for transfer learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 193–200.
[44] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning*. Cham, Switzerland: Springer, 2018, pp. 270–279.
[45] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, Dec. 2016.

[46] Z. Deng, K.-S. Choi, Y. Jiang, and S. Wang, "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2585–2599, Dec. 2014.

[47] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4133–4141.

[48] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, Mar. 2021.

[49] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 742–751.

[50] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5142–5151.

[51] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 4794–4802.

[52] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1542–1547.

[53] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[54] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. Ph.D. Workshop (IIPhDW)*, May 2018, pp. 117–122.

[55] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Netw. Vis. Recognit*, vol. 11, pp. 1–8, Dec. 2017.

[56] Y. Jiang, J. Xue, R. Wang, K. Xia, X. Gu, J. Zhu, L. Liu, and P. Qian, "Seizure recognition using a novel multitask radial basis function neural network," *J. Med. Imag. Health Informat.*, vol. 9, no. 9, pp. 1865–1870, Dec. 2019.

[57] S. Hashem, G. Esmat, W. Elakel, S. Habashy, S. A. Raouf, M. Elhefnawi, M. I. Eladawy, and M. ElHefnawi, "Comparison of machine learning approaches for prediction of advanced liver fibrosis in chronic hepatitis c patients," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 3, pp. 861–868, May 2017.

[58] G. Hoffmann, A. Bietenbeck, R. Lichtinghagen, and F. Klawonn, "Using machine learning techniques to generate laboratory diagnostic pathways-a case study," *J. Lab. Precis. Med.*, vol. 3, p. 58, Jun. 2018.

[59] L. Ralf, K. Frank, and H. Georg. (2017). *HCV Data Data Set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/HCV+data

[60] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml and http://archive.ics.uci.edu/ml/citation_policy.html

[61] D. Chicco and G. Jurman, "An ensemble learning approach for enhanced classification of patients with hepatitis and cirrhosis," *IEEE Access*, vol. 9, pp. 24485–24498, 2021.

[62] S. S. Chawathe, "Diagnostic classification using hepatitis c tests," in *Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS)*, Sep. 2020, pp. 1–7.

[63] M. M. Denniston, R. M. Klevens, G. M. McQuillan, and R. B. Jiles, "Awareness of infection, knowledge of hepatitis C, and medical follow-up among individuals testing positive for hepatitis C: National health and nutrition examination survey 2001–2008," *Hepatology*, vol. 55, no. 6, pp. 1652–1661, Jun. 2012.

[64] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," *Nature Med.*, vol. 25, no. 1, pp. 44–56, 2019.

[65] C. C. Serdar, M. Cihan, D. Yücel, and M. A. Serdar, "Sample size, power and effect size revisited: Simplified and practical approaches in pre-clinical, clinical and laboratory studies," *Biochemia Medica*, vol. 31, no. 1, pp. 27–53, Feb. 2021.

[66] E. M. Langberg, L. Dyhr, and A. S. Davidsen, "Development of the concept of patient-centredness—A systematic review," *Patient Educ. Counseling*, vol. 102, no. 7, pp. 1228–1236, Jul. 2019.

[67] "Exploratory data analysis," in *The Concise Encyclopedia of Statistics*. New York, NY, USA: Springer, 2008, pp. 192–194, doi: 10.1007/978-0-387-32833-1_136.

[68] J. T. Behrens, "Principles and procedures of exploratory data analysis," *Psychol. Methods*, vol. 2, no. 2, p. 131, 1997.

[69] T. Milo and A. Somech, "Automating exploratory data analysis via machine learning: An overview," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2020, pp. 2617–2622.

[70] K. Sahoo, A. K. Samal, J. Pramanik, and S. K. Pani, "Exploratory data analysis using Python," *Int. J. Innov. Technol. Exploring Eng.*, vol. 8, no. 12, pp. 1–9, 2019.

[71] A. Alwosheel, S. van Cranenburgh, and C. G. Chorus, "Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis," *J. Choice Model.*, vol. 28, pp. 167–182, Sep. 2018.

[72] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: When to warp?" in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2016, pp. 1–6.

[73] D. A. van Dyk and X.-L. Meng, "The art of data augmentation," *J. Comput. Graph. Statist.*, vol. 10, no. 1, pp. 1–50, 2001.

[74] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *J. Amer. Statist. Assoc.*, vol. 82, no. 398, pp. 528–540, 1987.

[75] Y. Dodge, D. Cox, and D. Commenges, *The Oxford Dictionary of Statistical Terms*. Oxford, U.K.: Oxford Univ. Press Demand, 2006.

[76] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM Rev.*, vol. 56, no. 1, pp. 3–69, 2014.

[77] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The Mahalanobis distance," *Chemometrics Intell. Lab. Syst.*, vol. 50, no. 1, pp. 1–18, 2000.

[78] T. Kløve, T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Permutation arrays under the Chebyshev distance," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2611–2617, Jun. 2010.

[79] P. J. F. Groenen and K. Jajuga, "Fuzzy clustering with squared Minkowski distances," *Fuzzy Sets Syst.*, vol. 120, no. 2, pp. 227–237, Jun. 2001.

[80] E. Choi and C. Lee, "Feature extraction based on the Bhattacharyya distance," *Pattern Recognit.*, vol. 36, no. 8, pp. 1703–1709, 2003.

[81] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jul. 2002.

[82] M. Y. Rafiq, G. Bugmann, and D. J. Easterbrook, "Neural network design for engineering applications," *Comput. Struct.*, vol. 79, no. 17, pp. 1541–1552, 2001.

[83] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[84] A. LeNail, "NN-SVG: Publication-ready neural network architecture schematics," *J. Open Source Softw.*, vol. 4, no. 33, p. 747, Jan. 2019.

[85] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[86] J. Larsen, L. K. Hansen, C. Svarer, and M. Ohlsson, "Design and regularization of neural networks: The optimal use of a validation set," in *Proc. IEEE Signal Process. Soc. Workshop*, Sep. 1996, pp. 62–71.

[87] Y. Xu and R. Goodacre, "On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning," *J. Anal. Test.*, vol. 2, no. 3, pp. 249–262, Jul. 2018.

[88] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, p. e623, Jul. 2021.

[89] W. Sun, "Stability of machine learning algorithms," Dept. Statist., Purdue Univ., West Lafayette, IN, USA, 2015.

[90] P. Turney, "Bias and the quantification of stability," *Mach. Learn.*, vol. 20, no. 1, pp. 23–33, 1995.

[91] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, "A comprehensive survey of loss functions in machine learning," *Ann. Data Sci.*, vol. 9, no. 2, pp. 187–212, Apr. 2022.

[92] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.

[93] T. Agrawal, "Optuna and AutoML," in *Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient*. Berkeley, CA, USA: Apress, 2021, pp. 109–129, doi: 10.1007/978-1-4842-6579-6_5.

[94] L. Prechelt, "Early stopping—But When?" in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 1998, pp. 55–69, doi: 10.1007/3-540-49430-8_3.

[95] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Netw.*, vol. 11, no. 4, pp. 761–767, 1998.

[96] W. McKinney, "Pandas: A foundational Python library for data analysis and statistics," *Python High Perform. Sci. Comput.*, vol. 14, no. 9, pp. 1–9, 2011.
[97] T. E. Oliphant, *A Guide to NumPy*. USA: Trelgol Publishing, 2006.
[98] M. Alabadla, F. Sidi, I. Ishak, H. Ibrahim, L. S. Affendey, Z. C. Ani, M. A. Jabar, U. A. Bukar, N. K. Devaraj, A. S. Muda, A. Tharek, N. Omar, and M. I. M. Jaya, "Systematic review of using machine learning in imputing missing values," *IEEE Access*, vol. 10, pp. 44483–44502, 2022.
[99] A. G. Asuero, A. Sayago, and A. González, "The correlation coefficient: An overview," *Crit. Rev. Anal. Chem.*, vol. 36, no. 1, pp. 41–59, 2006.
[100] M. L. Waskom, "Seaborn: Statistical data visualization," *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, 2021, doi: 10.21105/joss.03021.
[101] H.-H. Hsu and C.-W. Hsieh, "Feature selection via correlation coefficient clustering," *J. Softw.*, vol. 5, no. 12, pp. 1371–1377, Dec. 2010.
[102] B. Ratner, "The correlation coefficient: Its values range between +1/−1, or do they?" *J. Targeting, Meas. Anal. Marketing*, vol. 17, no. 2, pp. 139–142, Jun. 2009.
[103] S. Gupta and A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review," *Proc. Comput. Sci.*, vol. 161, pp. 466–474, Jan. 2019.
[104] A. Atla, R. Tada, V. Sheng, and N. Singireddy, "Sensitivity of different machine learning algorithms to noise," *J. Comput. Sci. Colleges*, vol. 26, no. 5, pp. 96–103, 2011.
[105] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, nos. 5–6, pp. 183–197, Jul. 1991.
[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[107] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "AutoGluon-tabular: Robust and accurate AutoML for structured data," 2020, *arXiv:2003.06505*.
[108] J. Mueller, X. Shi, and A. Smola, "Faster, simpler, more accurate: Practical automated machine learning with tabular, text, and image data," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 3509–3510.
[109] A. Ogunleye and Q.-G. Wang, "XGBoost model for chronic kidney disease diagnosis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 6, pp. 2131–2140, Nov. 2019.
[110] L. Torlay, M. Perrone-Bertolotti, E. Thomas, and M. Baciu, "Machine learning–XGBoost analysis of language networks to classify patients with epilepsy," *Brain Inf.*, vol. 4, no. 3, pp. 159–169, Sep. 2017.
[111] L. Sun, "Application and improvement of xgboost algorithm based on multiple parameter optimization strategy," in *Proc. 5th Int. Conf. Mech., Control Comput. Eng. (ICMCCE)*, Dec. 2020, pp. 1822–1825.
[112] S. Putatunda and K. Rama, "A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost," in *Proc. Int. Conf. Signal Process. Mach. Learn. (SPML)*, 2018, pp. 6–10.
[113] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized XGBoost based diagnostic system for effective prediction of heart disease," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4514–4523, Jul. 2020.
[114] E. A. Daoud, "Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset," *Int. J. Comput. Inf. Eng.*, vol. 13, no. 1, pp. 6–10, 2019.
[115] D. Wang, Y. Zhang, and Y. Zhao, "LightGBM: An effective miRNA classification method in breast cancer patients," in *Proc. Int. Conf. Comput. Biol. Bioinf. (ICCBB)*, 2017, pp. 7–11.
[116] T.-H.-S. Li, H.-J. Chiu, and P.-H. Kuo, "Hepatitis c virus detection model by using random forest, logistic-regression and ABC algorithm," *IEEE Access*, vol. 10, pp. 91045–91058, 2022.
[117] M. O. Edeh, S. Dalal, I. B. Dhaou, C. C. Agubosim, C. C. Umoke, N. E. Richard-Nnabu, and N. Dahiya, "Artificial intelligence-based ensemble learning model for prediction of hepatitis c disease," *Frontiers Public Health*, vol. 10, p. 847, Apr. 2022.

LERAN CHEN was born in Shandong, China, in 1998. He received the B.S. degree in mechanical engineering from the Southern University of Science and Technology (SUSTech), in 2020, where he is currently pursuing the Ph.D. degree with the joint Ph.D. degree between SUSTech and The Hong Kong Polytechnic University. His research interests include machine learning, custom algorithms, and smart manufacturing.

PING JI received the Ph.D. degree in USA. In 1984, he joined as an Assistant Lecturer at Beihang University. He was at the National University of Singapore (NUS), in 1992. He joined The Hong Kong Polytechnic University (PolyU), Hong Kong, in 1996, where he is currently a Professor with the Department of Industrial and Systems Engineering. He has authored or coauthored more than 100 journal articles. His current research interests include enterprise resources planning, operations management and optimization, and its applications.

YONGSHENG MA received the B.Eng. degree from Tsinghua University, Beijing, in 1986, and the M.Sc. and Ph.D. degrees from UMIST, U.K., in 1990 and 1994, respectively.

He started his career as a Polytechnic Lecturer in Singapore from 1993 to 1996; and then a Research Fellow, a Senior Research Fellow, and a Group Manager from 1996 to 2000 at the Singapore Institute of Manufacturing Technology. He was an Associate Professor with Nanyang Technological University, Singapore, from 2000 to 2007. He was a Full Professor at the University of Alberta (UA) from 2007 to 2021. He has been a Full Professor with the Southern University of Science and Technology (SUSTech), Shenzhen, China, since July 2021. He has an established research profile with many research projects from different sources, and published more than 200 papers internationally in recognized top journals, conferences, and book chapters. His research interests include CAx interoperability, CAD/CAE integration, collaborative and concurrent engineering in MRP/ERP/CRM, and product life cycle management. His specialty is in feature-based intelligent product and engineering process informatics.

Dr. Ma has been a member of ASEE, SME, SPE, ASME, CSME and a Canada (Alberta) registered Professional Engineer (P.Eng.), since 2009. In 2012, he received the prestigious ASTech Award from The Alberta Science and Technology Leadership Foundation together with Drader Manufacturing Ltd. He was an Associate Editor of IEEE TRANSACTIONS OF AUTOMATION SCIENCE AND ENGINEERING, from 2009 to 2013. He has been an Editorial Board Member of *Advanced Engineering Informatics* (ADVEI, Elsevier), since 2012, and has been an Associate Editor, since 2020. Concurrently, he is an Associate Editor of *ASME Journal of Computer Information Science and Engineering* (JCISE), and an Editorial Member of *Scientific Reports* (Springer Nature).