Journal of Future Artificial Intelligence and Technologies

E-ISSN: 3048-3719

Research Article

Analyzing Quantum Feature Engineering and Balancing Strategies Effect on Liver Disease Classification

Achmad Nuruddin Safriandono [1], De Rosal Ignatius Moses Setiadi [2,*], Akhmad Dahlan [3], Farah Zakiyah Rahmanti [4], Iwan Setiawan Wibisono [5], and Arnold Adimabua Ojugo [6]

[1] Department of Computer Systems Faculty of Engineering, Sultan Fatah University, Demak, Indonesia; e-mail : udinozz@gmail.com
[2] Department of Informatic Engineering, Faculty of Computer Science, Dian Nuswantoro University, Semarang, Indonesia; e-mail : moses@dsn.dinus.ac.id
[3] Department of Informatics Management, Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia; e-mail : alland@amikom.ac.id
[4] Department of Information Technology, School of Computing, Telkom University, Indonesia; e-mail : farahzakiyah@telkomuniversity.ac.id
[5] Informatics Engineering Department, Faculty of Computers and Education, Ngudi Waluyo University, Indonesia; e-mail : iwansetiawan@unw.ac.id
[6] Department of Computer Science, Federal University of Petroleum Resources Effurun, Delta State, Nigeria; e-mail : ojugo.arnold@fupre.edu.ng
* Corresponding Author : De Rosal Ignatius Moses Setiadi

Abstract: This research aims to improve the accuracy of liver disease classification using Quantum Feature Engineering (QFE) and the Synthetic Minority Over-sampling Tech-nique and Tomek Links (SMOTE-Tomek) data balancing technique. Four machine learning models were compared in this research, namely eXtreme Gradient Boosting (XGB), Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR) on the Indian Liver Patient Dataset (ILPD) dataset. QFE is applied to capture correlations and complex patterns in the data, while SMOTE-Tomek is used to address data imbalances. The results showed that QFE significantly improved LR performance in terms of recall and specificity up to 99%, which is very important in medical diagnosis. The combination of QFE and SMOTE-Tomek gives the best results for the XGB method with an accuracy of 81%, recall of 90%, and f1-score of 83%. This study concludes that the use of QFE and data balancing techniques can improve liver disease classification performance in general.

Keywords: Data augmentation; Feature selection; Hepatitis classification; Preprocessing effect; Quantum Feature Engineering.

Received: April, 24th 2024
Revised: May, 25th 2024
Accepted: May, 27th 2024
Published: June, 1st 2024
Curr. Ver.: July, 3rd 2024

1. Introduction

Liver disease or hepatitis is a significant global health problem caused by excessive alcohol consumption, hepatitis virus infection, and obesity. According to the World Health Organization (WHO)[1], hepatitis causes various health problems, including death. There are five main types of hepatitis viruses: A, B, C, D, and E, with types B and C being the main causes of chronic disease, liver cirrhosis, and liver cancer. An estimated 354 million people are living with hepatitis B or C. Vaccination and education campaigns could prevent an estimated 4.5 million premature deaths by 2030. Hepatitis classification research is important for early detection, accurate diagnosis, appropriate treatment, and the development of fairer algorithms, as well as addressing health inequalities[2]–[4].

Various studies have been conducted on disease classification using machine learning (ML). Several popular ML methods such as Random Forest(RF)[3], [5]–[8], Support Vector Machine (SVM)[3], [8]–[11], eXtreme Gradient Boosting(XGB)[7], [8], [11], [12], k-nearest Neighboor(KNN)[5], [9], [13], [14], Naïve Bayes(NB)[3], [8], [13], and Logistic Regression (LR)[7][5], [8], [11]. Looking at the data presented in[15], four ML methods have the best

performance for classifying hepatitis datasets, namely RF, SVM, XGB, and LR. Each classification method has advantages and disadvantages. RF excels at overfitting and handling imbalanced data but is complex and resource-intensive. SVM is effective in high-dimensional spaces and has good performance with maximum margins but is slow for large data and requires appropriate kernel selection. XGB is fast and efficient with good overfitting control but is complex and requires a lot of memory and training time[16]–[18]. LR is simple, easy to interpret and efficient for linear data, but it is limited in handling nonlinear relationships and prone to overfitting in high dimensions. The choice of algorithm depends on the characteristics of the dataset and the objectives of the analysis.

Several hepatitis datasets have different characteristics. One of the datasets, namely the Indian Liver Patient Dataset (ILPD), has a relatively smaller number of features[3]. Classifying datasets with few features presents several significant challenges, such as limited information, overfitting, limited model complexity, model bias, difficulty in feature selection, and lack of variation. Limited information can reduce a model's ability to make accurate predictions, and models tend to learn irrelevant patterns, leading to overfitting. More complex methods may be ineffective, and models can be highly biased if the available features are not representative enough. Feature Engineering is one solution to deal with this[19]–[22]. The way it works is to create new features from existing features to increase the available information and help the model make more accurate predictions. However, in research[23], the use of feature selection contributed to an increase in accuracy of 2%.

Popular Feature Engineering includes Scaling, Encoding, Interaction Features, and Polynomial Features. Scaling rescales data using normalization or standardization to keep all features within the same value range, helping models work more effectively with balanced data. Encoding converts category features into a numeric format using methods such as One-Hot Encoding or Label Encoding, allowing the model to understand and process category features. Interaction Features involve creating new features based on interactions between existing features to capture more complex nonlinear relationships. Polynomial Features create new features from polynomial combinations of original features, allowing the model to capture nonlinear relationships that increase prediction accuracy[24]–[26]. Quantum Feature Engineering (QFE) is a new approach that uses quantum computing principles for feature transformation. By using qubits that can be in a superposition of many states at once, QFE can capture more complex correlations and patterns in data, offering increased computational efficiency and the ability to solve problems that are difficult to address with classical methods, giving ML models increased performance in large and complex datasets [27]–[29].

Another problem in various datasets, especially in medical datasets, is the imbalanced number of records in each class. Imbalanced data occurs when the number of samples from the majority class is much greater than that from the minority class. This can cause the ML model to be biased towards the majority class and ignore the minority class, which is often more important to identify[30]. To overcome this problem, data balancing techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and Tomek Links are used. SMOTE increases the number of minority class samples by creating new synthetic samples[31], [32], while Tomek Links removes adjacent samples from the majority and minority classes that are considered difficult to classify correctly[33]. The combination of SMOTE and Tomek Links (SMOTE-Tomek) can help reduce bias and improve model performance in detecting cases from minority classes, which is crucial in medical applications to ensure correct and fair diagnosis for all patients[18]. Based on this literature, this research aims to:

1. Four ML models, namely XGB, RF, SVM, and LR, were compared to classify liver disease in the LIPD dataset.
2. Analyze the effects of using QFE on the four selected models.
3. Integrating SMOTE-Tomek and QFE to get the best performance in ML models.

The rest of the paper is organized into five sections. Section 2 reviews related literature that uses the same dataset and highlights the research gaps. Section 3 details the proposed method and its stages. Section 4 presents the results and compares the three preprocessing approaches. Section 5 discusses the comparisons and concludes the paper.

2. Related Works

Several previous studies have developed and analyzed classification methods for detecting liver disease or hepatitis, such as in research [13] which tested several methods such as

KNN, NB, and C.4.5 to classify the ILPD dataset and the 'new' dataset (NDS) created by the author the paper. The performance of the three methods is quite competitive on each matrix for the NDS and ILPD datasets. For accuracy on the ILPD C.4.5 dataset, it is significantly superior, whereas it is significantly weaker in precision and specificity. For the ILPD dataset, the best accuracy is 0.69 for the C4.5 method, recall is 0.75 for the NB method, precision is 0.78 for the KNN method, and specificity is 0.98 for the KNN method. Research [34] also compared several methods for classifying medical datasets, specifically on the ILPD dataset. The best accuracy was 0.69 for the J48 method, the best recall for the NB method was 0.95, the best precision for the bagging method was 0.44, and the best specificity for the bagging method was 0.86.

Another research in [23] compared several methods, such as LR, Sequential Minimal Optimization (SMO), RF, NB, J48, and KNN, on the IPLD dataset. Comparison of the results of various algorithms is carried out with and without feature selection techniques. After the feature selection process, the features selected are Total Bilirubin, Direct Bilirubin, Alkaline Phosphatase, Serum Glutamic Pyruvic Transaminase, and Serum Glutamic Oxaloacetic Transaminase. LR is the method with the best accuracy performance, LR also experienced the highest increase in accuracy after feature selection, from 72.50% to 74.36%. Other methods that experienced an increase in performance after feature selection were RF and J48, while other methods decreased and were relatively stagnant.

Research [3] analyzes the ILPD dataset from a different perspective, focusing on differences based on gender features. Several classification models, such as RF, SVM, NB, and LR, in four different experiments. The results generally show significant performance differences between the genders, with women tending to experience higher false negative rates than men. This means that women are more likely to have undetected liver disease, which can lead to a lack of appropriate treatment. In more detail, RF and LR significantly differ in false negative rate (FNR) between genders, where RF shows an FNR of -21.02% for women, while LR is -24.07%. Experiments also show that using a gender-balanced dataset can improve overall performance: RF increased from 78.17% (SD 2.36) to 81.66% (SD 2.33), LR increased from 71.31% (SD 2.37) to 74.53% (SD 1.96), SVM increased from 79.40% (SD 2.50) to 83.30% (SD 1.75), and NB increased from 71.53% (SD 2.61) to 74.75% (SD 1.9).

Research [35] focuses more on developing the KNN method with Variable-Neighbor Weighted Fuzzy (Variable-NWFKNN) combined with Tomek Link and Reredundancy-based Undersampling (TL_RUS) to be tested on three liver disease datasets, namely ILPD, Madhya Pradesh Region Liver Patient Dataset (MPRLPD) and BUPA Medical Research Ltd. (BUPA). TL_RUS functions to balance the dataset with undersampling techniques. Variable-NWFKNN improves classification performance on imbalanced datasets by giving greater weight to neighbors from minority classes and varying the number of neighbors (k) to overcome zero division error. This method combines normalization, standardization, and preprocessing with TL RUS to reduce bias and consistently improve classification performance on all three datasets. Specifically, the ILPD dataset resulted in an accuracy of 87.71%, sensitivity of 90.03%, specificity of 74.79%, precision of 95.21%, and f1-score of 92.55%.

Based on several related works, it can be concluded that several studies only compared several standard methods. Then, several more specific analyses were carried out on balancing based on gender features. Furthermore, optimization of the classification method and several preprocessing techniques are also carried out, which are crucial to the final performance. This research analyzes other techniques more deeply, such as preprocessing based on QFE and the SMOTE-Tomek over-under sampling combination.

3. Proposed Method

3.1. Dataset Collection

This research uses the ILPD dataset, which can be downloaded at https://archive.ics.uci.edu/dataset/225/ilpd+indian+liver+patient+dataset. This dataset has 583 instances, ten features, and multivariate characteristics. Of the 583 records, 416 patients were diagnosed with liver disease and 167 patients without liver disease. It was also known that 441 were male and 142 were female. More detailed features are in Table 1.

Table 1. Features details on ILPD dataset.

No	Feature	Data Type	Description
1	Age	Int	Age of the patient in years
2	Gender	Categorical	Gender of the patient (Male/Female)
3	Total_Bilirubin	Float	Total bilirubin level in blood (mg/dL)
4	Direct_Bilirubin	Float	Direct bilirubin level in blood (mg/dL)
5	Alkaline_Phosphotase	Int	Alkaline phosphatase enzyme level in blood (IU/L)
6	Alamine_Aminotransferase	Int	Alanine aminotransferase enzyme level in blood (IU/L)
7	Aspartate_Aminotransferase	Int	Aspartate aminotransferase enzyme level in blood (IU/L)
8	Total_Proteins	Float	Total protein level in blood (g/dL)
9	Albumin	Float	Albumin level in blood (g/dL)
10	Albumin_and_Globulin_Ratio	Float	Ratio of albumin to globulin in blood
11	Outcome	Int	Diagnosis result (1: Liver Patient, 2: Non-Liver Patient)

3.2. Preprocessing

In this research, three main preprocessing stages are presented, along with three tests. The first test was carried out using first-stage preprocessing, and the second test used first- and second-stage preprocessing. Finally, the third test was carried out in the order of first, third, and second stage preprocessing. Training and model evaluation are carried out after preprocessing is carried out according to these provisions. Each preprocessing stage is explained as follows.

3.2.1. First Stage

At this stage, duplicate records and missing values are first removed to ensure data quality and integrity. Duplicate data can cause the model to learn unrepresentative patterns and increase bias, while missing values can result in analysis errors and reduce model performance. Second, label encoding on the gender feature converts text category data such as "male" and "female" into a numeric format, for example, 0 and 1. This facilitates processing by machine learning algorithms, saves memory and resources, and simplifies data analysis. Third, change the target/outcome labels from [1, 2] to [0, 1] to match the binary format commonly used in classification (see Figure 2). Then, in the first stage's final step, the features (X) from the target (y) in the dataset are separated. The dataset in the first stage will be tested for classification in the machine learning model proposed in section 3.3.

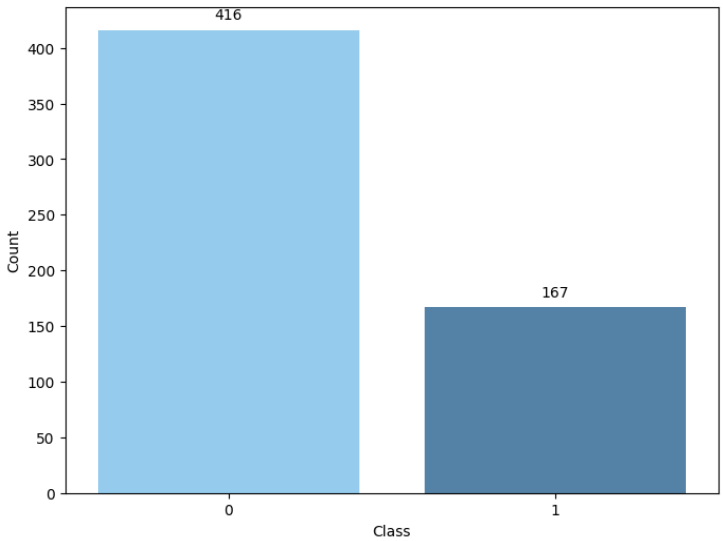


Figure 2. Class distribution ILPD Dataset, 0 for a liver patient, 1 for a non-liver patient.

3.2.2. Second Stage – Quantum Feature Engineering (QFE)

QFE is used to improve the performance of classification models through the use of quantum computing. QFE maps classical data into a quantum feature space, capturing data complexity that classical methods cannot reach. This process begins by converting bits into qubits and then feeding them to a Hadamard layer, where each qubit is mapped into a superposition by applying a Hadamard (H) gate, expanding the search space and enabling the exploration of more quantum configurations. Hadamard gate is expressed by Equation (1).

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1}$$

It then proceeds with a parametric rotation layer consisting of three gates (RX, RY, RZ) applied to each qubit based on the classical input values, which helps in encoding classical data into quantum space. This rotation gate is expressed in Equation (2)-(4).

$$RX(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{2}$$

$$RY(\theta) = \begin{pmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{pmatrix} \tag{3}$$

$$RZ(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{4}$$

The final layer of entanglement using CNOT gates creates complex quantum correlations between qubits, allowing the model to capture non-linear dependencies better. The CNOT gate is expressed in Equation (5).

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5}$$

The rotation and CNOT layers were performed three times to obtain better feature mapping. After quantum feature mapping, these features are scaled back to classical bit form then normalized to ensure consistent scaling before being used in classical model training, such as Logistic Regression. This combination of techniques produces a richer data representation and increases the model's ability to predict outcomes. The circuit visualization in the proposed QFE process is presented in Figure 3.

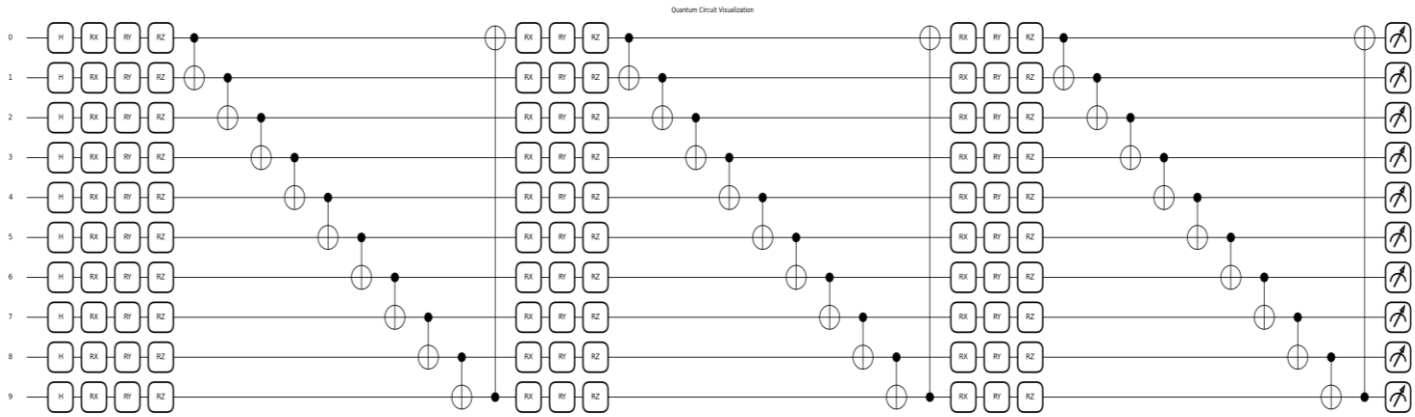


Figure 3. Proposed QFE visualization.

Next, the resulting quantum features are normalized with StandardScaler from scikit-learn. Normalization is done by subtracting the original and mean values and then dividing it by the standard deviation, resulting in a distribution with a mean of 0 and a standard deviation of 1. Equation (6) is used to carry out the standardization process.

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

Where z is the standardized feature value, x is the original feature value, μ is the mean of the feature, and σ is the standard deviation of the feature. Standardization is applied to training and test data to ensure that all features are at the same scale, which is important for the optimal performance of many machine learning algorithms.

3.2.3. Third Stage –SMOTE-Tomek

Third stage preprocessing is carried out for the third test, namely testing with data balance and with QFE. Applying SMOTE-Tomek first ensures that the dataset is balanced, which is critical because class imbalance can significantly impact the performance of a machine learning model. Balancing the dataset before applying any feature engineering helps ensure that subsequent steps operate on a more representative sample of the data. In SMOTE, a new point is created by taking two nearby minority samples and generating a new sample between them. The SMOTE interpolation formula is shown in Equation (7).

$$x_{new} = x_i + (x_j - x_i) \times \lambda \tag{7}$$

Where x_i, x_j is a nearby minority sample, and λ is a random value between 0 and 1. After applying SMOTE, the Tomek Links technique is used to clear decision boundaries between classes by removing pairs of samples that are close together but come from different classes, thereby reducing overlap and increasing the clarity of class boundaries. A combination of SMOTE and Tomek Links, known as SMOTE-Tomek, is applied only to the training data to ensure a more balanced dataset.

After data balancing, the dataset is split into training and test sets using the train_test_split function of scikit-learn. The training set is used to build the model, while the testing set is used to evaluate the model's performance. The next step is feature normalization with a standard scaler with Equation (6). Finally, before entering the training stage, Recursive Feature Elimination (RFE) is used to select the features that will be carried out QFE. In the third stage, not all features are QFEed, but only four features selected by RFE are QFEed in section 3.2.2. Next, the four features carried out by QFE are combined with classical features and classified.

3.3. Machine Learning Models Configuration

The classification model was built using classes in the scikit-learn library. As a limitation, the configuration of each ML model is only simple without a grid search process to find the best hyperparameters, which is done in more detail on the effects of preprocessing. A simple configuration of each ML model can be seen in Table 2.

Table 2. Machine learning model configuration.

Model	Configuration
XGB	random_state=42
RF	n_estimators=150, random_state=42
LR	random_state=42
SVM	probability=True, random_state=42

3.4. Evaluation

The performance of all classification models is evaluated using several metrics that can be calculated using the confusion matrix, namely accuracy, recall, precision, specificity, and f1 score. A confusion matrix is a table used to describe the performance of a classification model. It summarizes the outcomes of predictions by comparing the actual labels with the predicted labels. In the context of the classification of liver disease, Matix confusion is visualized in Figure 4.

Accuracy represents the proportion of correctly predicted instances out of the total instances, providing a general overview of the model's performance for both classes. Conversely, Recall quantifies the percentage of true positive cases (liver patients) that the model successfully identifies. For liver disease detection, recall reflects the model's capability to

detect patients with liver disease accurately. Precision measures the percentage of positive predictions (predicted liver patients) that are actually correct. In the context of liver disease, precision reveals how many of the patients predicted to have liver disease genuinely have the condition. Specificity calculates the percentage of true negative cases (non-liver patients) that the model correctly identifies. For liver disease, specificity shows the model's effectiveness in recognizing patients who do not have liver disease. Lastly, the F1 score, which is the harmonic mean of precision and recall, offers a balanced measure that is particularly useful when dealing with the trade-offs between false positives and false negatives. The mathematical formulas for accuracy, recall, precision, speficity, and f1-score metrics are expressed in Equation (8)-(12), respectively.

	Predicted Positive (Liver Patient)	Predicted Negative (Non-Liver Patient)
Actual Positive (Liver Patient)	True Positive (TP)	False Negative (FN)
Actual Negative (Non-Liver Patient)	False Positive (FP)	True Negative (TN)

Figure 4. Confusion matrix for liver disease classification.

$$\text{acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$\text{rec} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{prec} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{spec} = \frac{TN}{TN + FP} \tag{11}$$

$$\text{f1} = \frac{2 \cdot (\text{prec} \cdot \text{rec})}{\text{prec} + \text{rec}} \tag{12}$$

TP: A patient who has liver disease is correctly diagnosed as having liver disease; TN: A patient who does not have liver disease is correctly diagnosed as not having liver disease; FP: A patient who does not have liver disease is incorrectly diagnosed as having liver disease; FN: A patient who has liver disease is incorrectly diagnosed as not having liver disease.

4. Results and Discussion

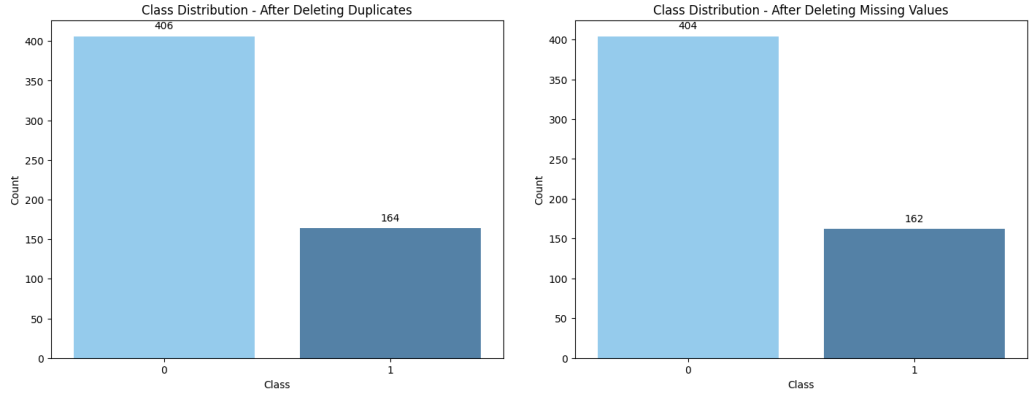
This research was implemented using Google Collab as the editor and Python as the programming language. This section is explained into two large parts, namely, three stages of dataset preprocessing and a discussion of classification results.

4.1. First Stage Preprocessing

The first preprocessing stage is done by deleting duplicate data and records with missing values. There are ten duplicate data in the liver patient class and three duplicate data in non-liver patients. The results of the duplicate removal process are shown in Figure 5(a). Meanwhile, to remove missing values, 2 records were found in each class; the results are shown in Figure 5(b).

Next, the process of normalizing values is carried out using a standard scaler and separating features and targets. The sample dataset before normalization is shown in Figure 6, and after normalization is shown in Figure 7. Col_0, Col_1, ... Col_9 shows the ten features in

Table 1 in the same order. The preprocessing results at this stage are then tested for classification to determine the plain performance of the model without QFE and SMOTE-Tomek.



(a) (b)

Figure 5. Class distribution (**a**) after deleting duplicates; (**b**) after deleting missing values.

	COL_0	COL_1	COL_2	COL_3	COL_4	COL_5	COL_6	COL_7	COL_8	COL_9	Outcome
0	62	1	10.9	5.5	699	64	100	7.5	3.2	0.74	0
1	62	1	7.3	4.1	490	60	68	7.0	3.3	0.89	0
2	58	1	1.0	0.4	182	14	20	6.8	3.4	1.00	0
3	72	1	3.9	2.0	195	27	59	7.3	2.4	0.40	0
4	46	1	1.8	0.7	208	19	14	7.6	4.4	1.30	0

Figure 6. Sample dataset before normalization and have target field.

	COL_0	COL_1	COL_2	COL_3	COL_4	COL_5	COL_6	COL_7	COL_8	COL_9
0	0.252552	-1.742330	-0.394017	-0.456338	-0.563817	-0.340779	-0.323582	0.101543	0.574239	0.793631
1	0.931304	0.573944	-0.242755	-0.352774	-0.381151	-0.193081	-0.288854	-2.333100	-1.754166	-0.754717
2	-1.166655	0.573944	-0.318386	-0.387296	-0.500097	-0.232846	-0.160358	1.363951	1.064429	0.174292
3	1.054713	0.573944	1.133727	1.373292	1.734384	-0.085148	-0.028390	0.913091	0.084048	-0.630849
4	0.684485	0.573944	-0.363765	-0.421817	-0.406639	-0.323737	-0.278435	-0.619833	-0.651237	-0.445047

Figure 7. Sample dataset after normalization without target field.

4.2. Second Stage Preprocessing

At an early stage, QFE is carried out as described in stage 3.2.2. Ten classic features are processed with QFE, and the output from this stage is also 10 QFE features, as shown in Figure 8. Before these features are classified, normalization is repeated with a standard scaler, the results of which are shown in Figure 9.

	QFE_0	QFE_1	QFE_2	QFE_3	QFE_4	QFE_5	QFE_6	QFE_7	QFE_8	QFE_9
0	-0.068291	-0.008199	-0.021599	-0.008146	-0.019392	-0.004614	-0.038920	-0.060069	-0.067643	-0.015420
1	0.011154	0.079947	-0.018073	0.008530	0.002463	-0.035907	0.024375	0.028159	0.005344	-0.030358
2	0.000439	0.024851	0.029266	0.021173	0.025885	0.015806	0.035644	0.042503	0.000392	0.034939
3	-0.119957	0.048259	0.044314	0.042252	-0.078922	-0.011096	0.006764	-0.023192	0.083538	0.162491
4	0.068192	0.117639	-0.042077	0.012473	-0.011092	-0.007305	-0.003700	-0.017088	-0.059508	-0.054921

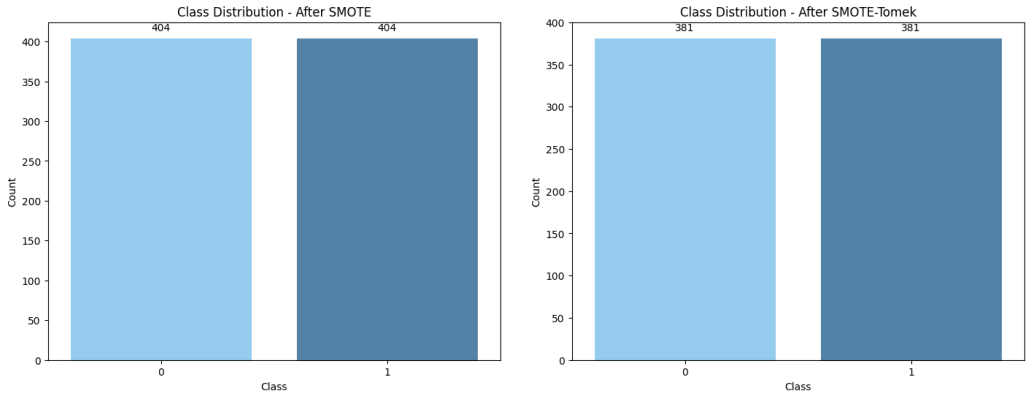
Figure 8. Sample dataset with QFE features.

	QFE_0	QFE_1	QFE_2	QFE_3	QFE_4	QFE_5	QFE_6	QFE_7	QFE_8	QFE_9
0	-2.079900	-0.690034	-0.608202	-0.581718	-0.768677	-0.405290	-0.944060	-1.459210	-1.495463	-0.428786
1	0.248379	0.801953	-0.533264	-0.079096	-0.161608	-1.300420	0.261688	0.441330	0.033013	-0.736207
2	-0.065622	-0.130612	0.473050	0.301978	0.489019	0.178837	0.476348	0.750317	-0.070709	0.607539
3	-3.594054	0.265603	0.792933	0.937346	-2.422315	-0.590709	-0.073792	-0.664847	1.670522	3.232406
4	1.919982	1.439933	-1.043526	0.039758	-0.538133	-0.482265	-0.273140	-0.533353	-1.325113	-1.241669

Figure 9. Sample dataset with QFE features after normalization.

4.3. Third Stage Preprocessing

At this stage, SMOTE-Tomek is carried out with input not from the second stage of preprocessing, but from the first preprocessing. So this third preprocessing is not carried out in the sequence process but the same process is repeated until the first preprocessing is used as input for this stage. SMOTE-Tomek is carried out with classical feature input. SMOTE was carried out to carry out oversampling of the minor class (non-liver patients). After this SMOTE process the number of minor and major classes became balanced as shown in Figure 10 (a). Then Tomek links are carried out to delete data that is considered to be intersecting to reduce noise and bias in the dataset. Each class has 404 records, after the Tomek process deleted 23 records for each, so that the remaining are 381 records for each class (see Figure 10 (b) which has 10 features.



(a) (b)

Figure 10. Sample dataset (**a**) After SMOTE; (**b**) After SMOTE-Tomek.

Next, the QFE process is carried out with the same circuit, but QFE is not on all features. The 4 best features were selected and carried out with RFE, then these four features were carried out with QFE. A sample of this stage is shown in Figure 11. Then, normalization is carried out with a standard scaler on the four QFE features, the sample results of which are presented in Figure 12. The final step in this stage is to combine the 10 classic features with the 4 normalized QFE features shown in Figure 13.

	QFE_0	QFE_1	QFE_2	QFE_3
0	0.554318	0.452201	0.439539	0.533983
1	0.391359	-0.183139	-0.378849	0.193680
2	0.051959	-0.023341	0.911719	-0.015119
3	0.458568	0.399357	0.391997	0.473545
4	-0.056457	-0.366461	0.035244	0.046079

Figure 11. Sample selected QFE features.

	QFE_0	QFE_1	QFE_2	QFE_3
0	0.911136	1.148498	0.919251	1.047278
1	0.302473	-1.893158	-2.936206	-0.209242
2	-0.965218	-1.128137	3.143711	-0.980202
3	0.553503	0.895509	0.695278	0.824121
4	-1.370159	-2.770806	-0.985397	-0.754239

Figure 12. Sample selected QFE features after normalization.

	COL_0	COL_1	COL_2	COL_3	COL_4	COL_5	COL_6	COL_7	COL_8	COL_9	QFE_0	QFE_1	QFE_2	QFE_3
0	0.746736	0.684543	-0.314066	-0.360394	-0.380603	-0.316511	-0.256781	-1.133839	-0.323622	0.759360	0.911136	1.148498	0.919251	1.047278
1	0.203800	0.684543	0.381746	0.548574	1.246444	1.567426	0.421715	0.018991	0.152814	0.110363	0.302473	-1.893158	-2.936206	-0.209242
2	0.203800	0.684543	-0.204194	-0.078632	1.391952	0.074495	0.110738	0.783641	-1.030222	-1.932515	-0.965218	-1.128137	3.143711	-0.980202
3	0.626083	-1.460829	-0.342062	-0.392235	-0.208639	-0.352057	-0.252742	-0.363334	0.152814	0.791322	0.553503	0.895509	0.695278	0.824121
4	1.530977	0.684543	-0.376529	-0.431435	2.600112	0.088713	0.381328	-0.172171	-0.635877	-0.740836	-1.370159	-2.770806	-0.985397	-0.754239

Figure 12. The final feature in the third stage preprocessing data.

4.4. Classification results

At this stage, three preprocessing methods were compared, and the results are presented in Tables 3, 4, and 5. Table 3 presents the results with input first stage preprocessing, consisting of deleting duplicates, deleting missing values, and normalizing processes. The result is that the SVM method performs best with accuracy, recall, f1, and specificity advantages. The RF and The LR methods perform worst with input first-stage preprocessing.

On the other hand, with input in the second stage, preprocessing LR is the best (see Table 4). QFE input contributes positively to boosting LR performance; even though the accuracy is relatively low, recall and specificity are the best and reach 0.99. These two measuring tools are the most important in the case of medical data because they determine the diagnosis results the most[10], [36]–[38]. XGB follows this with superior accuracy and precision and has the same f1 as LR. Next is the RF method, followed by the SVM method. The performance of the SVM method with QFE input is relatively similar to the standard classic feature input.

Table 3. Machine Learning models performance without QFE and SMOTE-Tomeks.

Model	Accuracy	Precision	Recall	F1 Score	Specificity
LR	0.68	0.74	0.88	0.80	0.88
SVM	**0.72**	0.75	**0.92**	**0.83**	**0.92**
RF	0.69	0.77	0.82	0.80	0.82
XGB	0.69	**0.78**	0.81	0.79	0.81

In the third experiment (see Table 5) with third-stage preprocessing input, XGB was the best, where XGB was almost superior in all metrics except precision. This result probably compares best with the XGB results in the two previous experiments, which were always superior in precision. However, using SMOTE-Tomek combined with QFE was no better than the second experiment (only using QFE in the LR method. The most significant increase was in accuracy, but the accuracy metric is actually not the main thing in medical cases. The performance of the LR and SVM methods dropped when compared The second and first experiments. The RF method also experienced a slight decrease compared to the second experiment but was better than the first experiment.

From the three preprocessing stages carried out, it can be seen that the QFE method contributes significantly to improving model performance, especially in recall and specificity, which are very important in medical applications. However, the combination of QFE and SMOTE-Tomek does not always provide the best results compared to using QFE alone. Choosing the right preprocessing technique greatly influences the final model results, and it

is important to consider metrics relevant to the application context, such as recall and specificity in medical diagnosis.

Table 4. Machine Learning models performance with QFE and without SMOTE-Tomeks.

Model	Accuracy	Precision	Recall	F1 Score	Specificity
LR	0.74	0.75	**0.99**	**0.85**	**0.99**
SVM	0.74	0.78	0.92	0.84	0.92
RF	**0.77**	0.79	0.94	0.86	0.94
XGB	**0.77**	**0.80**	0.92	**0.85**	0.92

Table 4. Machine Learning models performance with QFE and SMOTE-Tomeks.

Model	Accuracy	Precision	Recall	F1 Score	Specificity
LR	0.70	0.78	0.56	0.65	0.56
SVM	0.75	**0.84**	0.61	0.71	0.61
RF	0.78	0.83	0.69	0.75	0.69
XGB	**0.81**	0.77	**0.90**	**0.83**	**0.72**

5. Comparison

In this section, a comparison of the performance of the proposed model with other existing methods in the literature is carried out. Table 6 presents a comparison between the results of this study and several previous studies.

Table 6. Comparison with another method.

Method	Accuracy	Precision	Recall	F1 Score	Specificity
Ref [13]	0.69	0.69	0.74	-	0.76
Ref [34]	0.69	0.44	0.29	-	0.86
Ref [23]	0.74	-	-	-	-
Ref [35] balance	**0.88**	**0.95**	0.90	**0.92**	0.74
Ref [3] balance	0.83	0.78	0.92	0.84	0.92
Our (balance, QFE)	0.81	0.77	0.90	0.83	0.72
Ref [35] imbalance	0.78	0.89	0.82	0.85	0.59
Ref [3] imbalance	0.79	0.75	0.92	0.83	0.92
Our (imbalance)	0.72	0.75	0.92	0.83	0.92
Our (imbalance, QFE)	0.74	0.75	**0.99**	0.85	**0.99**

This study's results show that using QFE and data balancing techniques such as SMOTE-Tomek provides significant performance improvements in machine learning models for liver disease classification. On a balanced dataset, the method [35] has the highest accuracy of 0.88 with a precision of 0.95, recall of 0.90, and f1 score of 0.92. However, the specificity of this method is lower compared to other values. On an unbalanced dataset, this research shows recall up to 0.99 and a specificity of 0.99 using QFE. This shows that QFE can help improve model performance in detecting true positive and negative cases accurately[[10], [36]–[38]. This research shows that the use of Quantum Feature Engineering (QFE) can improve the performance of machine learning models in liver disease classification, especially in terms of recall and specificity, which are very important in medical applications. Additionally, data balancing techniques such as SMOTE-Tomek also contribute positively to model performance, although they do not always provide significant improvements in all metrics

6. Conclusions

This research shows that applying QFE and SMOTE-Tomek data balancing techniques can improve the performance of machine learning models in liver disease classification. QFE contributes to increased recall and specificity, which is important for accurate medical diagnosis. The XGB model performs best with 81% accuracy, 90% recall, and 83% f1-score on a

balanced dataset. This research indicates that the use of QFE can capture more complex patterns in the data, while SMOTE-Tomek helps reduce bias towards the majority class. However, combining QFE and SMOTE-Tomek does not always improve performance in all metrics. Therefore, this study suggests further exploration of additional techniques such as hybrid quantum-classical methods, more sophisticated data balancing techniques, and QFE applications on various other types of medical datasets for better generalization of the results. Further research is needed to understand more deeply the impact of these techniques on a wider range of dataset conditions and classification methods.

Author Contributions: Conceptualization: A.N.S and D.R.I.M.S; Methodology: A.N.S and D.R.I.M.S.; Software: A.D. and F.Z.R..; Validation: A.N.S, D.R.I.M.S, A.D. and F.Z.R..; Formal analysis: I.S.W.; Investigation: A.A.O.; Resources: A.N.S.; Data curation: A.D.; Writing—original draft preparation: A.N.S.; Writing—review and editing: D.R.I.M.S. and A.A.O, ; Visualization: F.Z.R.; Supervision: D.R.I.M.S.; Project administration: F.Z.R..; Funding acquisition: All.

Funding: This research received no external funding.

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

References

[1] WHO, "Hepatitis," *who.int*, 2024. https://www.who.int/health-topics/hepatitis#tab=tab_1 (accessed May 19, 2024).
[2] B. V. Ramana, P. M. Surendra, P. Babu, and P. N. B. Venkateswarlu, "A Critical Comparative Study of Liver Patients from USA and INDIA: An Exploratory Analysis," *Int. J. Comput. Sci. Issues*, vol. 9, no. 3, pp. 506–516, 2012.
[3] I. Straw and H. Wu, "Investigating for bias in healthcare algorithms: a sex-stratified analysis of supervised machine learning models in liver disease prediction," *BMJ Heal. Care Informatics*, vol. 29, no. 1, p. e100457, Apr. 2022, doi: 10.1136/bmjhci-2021-100457.
[4] M. B. Butt *et al.*, "Diagnosing the Stage of Hepatitis C Using Machine Learning," *J. Healthc. Eng.*, vol. 2021, pp. 1–8, Dec. 2021, doi: 10.1155/2021/8062410.
[5] H. Mamdouh Farghaly, M. Y. Shams, and T. Abd El-Hafeez, "Hepatitis C Virus prediction based on machine learning framework: a real-world case study in Egypt," *Knowl. Inf. Syst.*, vol. 65, no. 6, pp. 2595–2617, Jun. 2023, doi: 10.1007/s10115-023-01851-4.
[6] F. Mustofa, A. N. Safriandono, A. R. Muslikh, and D. R. I. M. Setiadi, "Dataset and Feature Analysis for Diabetes Mellitus Classification using Random Forest," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 41–48, Jan. 2023, doi: 10.33633/jcta.v1i1.9190.
[7] H. Ding, M. Fawad, X. Xu, and B. Hu, "A framework for identification and classification of liver diseases based on machine learning algorithms," *Front. Oncol.*, vol. 12, no. October, pp. 1–7, Oct. 2022, doi: 10.3389/fonc.2022.1048348.
[8] T. R. Noviandy, K. Nisa, G. M. Idroes, I. Hardi, and N. R. Sasmita, "Classifying Beta-Secretase 1 Inhibitor Activity for Alzheimer's Drug Discovery with LightGBM," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 358–367, Mar. 2024, doi: 10.62411/jcta.10129.
[9] V. K. Yarasuri, G. K. Indukuri, and A. K. Nair, "Prediction of Hepatitis Disease Using Machine Learning Technique," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Dec. 2019, pp. 265–269. doi: 10.1109/I-SMAC47947.2019.9032585.
[10] F. S. Gomiasti, W. Warto, E. Kartikadarma, J. Gondohanindijo, and D. R. I. M. Setiadi, "Enhancing Lung Cancer Classification Effectiveness Through Hyperparameter-Tuned Support Vector Machine," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 396–406, Mar. 2024, doi: 10.62411/jcta.10106.
[11] K. R. Singh, R. Gupta, R. K. Kadian, and R. Singh, "An Optimized XGBoost approach for Predicting Progression of Hepatitis C using Hyperparameter Tuning and Feature Interaction Constraint," in *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, Aug. 2022, pp. 1–8. doi: 10.1109/ASIANCON55314.2022.9909086.
[12] L. Ma, Y. Yang, X. Ge, Y. Wan, and X. Sang, "Prediction of disease progression of chronic hepatitis C based on XGBoost algorithm," in *2020 International Conference on Robots & Intelligent System (ICRIS)*, Nov. 2020, pp. 598–601. doi: 10.1109/ICRIS52159.2020.00151.
[13] M. S. P. Babu, M. Ramjee, S. Katta, and Swapna K, "Implementation of partitional clustering on ILPD dataset to predict liver disorders," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Aug. 2016, vol. 0, pp. 1094–1097. doi: 10.1109/ICSESS.2016.7883256.
[14] M. A. Araaf, K. Nugroho, and D. R. I. M. Setiadi, "Comprehensive Analysis and Classification of Skin Diseases based on Image Texture Features using K-Nearest Neighbors Algorithm," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 31–40, Sep. 2023, doi: 10.33633/jcta.v1i1.9185.
[15] UCI Machine Learning, "Hepatitis," 1988. https://archive.ics.uci.edu/dataset/46/hepatitis
[16] F. Omoruwou, A. A. Ojugo, and S. E. Ilodigwe, "Strategic Feature Selection for Enhanced Scorch Prediction in Flexible Polyurethane Form Manufacturing," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 346–357, Feb. 2024, doi: 10.62411/jcta.9539.
[17] E. B. Wijayanti, D. R. I. M. Setiadi, and B. H. Setyoko, "Dataset Analysis and Feature Characteristics to Predict Rice Production based on eXtreme Gradient Boosting," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 299–310, Feb. 2024, doi: 10.62411/jcta.10057.
[18] D. R. I. M. Setiadi, K. Nugroho, A. R. Muslikh, S. Wahyu, and A. A. Ojugo, "Integrating SMOTE-Tomek and Fusion Learning with XGBoost Meta-Learner for Robust Diabetes Recognition," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 23–38, 2024.

[19] Y. Zhang and Z. Wang, "Feature Engineering and Model Optimization Based Classification Method for Network Intrusion Detection," *Appl. Sci.*, vol. 13, no. 16, p. 9363, Aug. 2023, doi: 10.3390/app13169363.
[20] M. Oudah and A. Henschel, "Taxonomy-aware feature engineering for microbiome classification," *BMC Bioinformatics*, vol. 19, no. 1, p. 227, Dec. 2018, doi: 10.1186/s12859-018-2205-3.
[21] D. Dai *et al.*, "Using machine learning and feature engineering to characterize limited material datasets of high-entropy alloys," *Comput. Mater. Sci.*, vol. 175, p. 109618, Apr. 2020, doi: 10.1016/j.commatsci.2020.109618.
[22] M. Aamir and S. M. A. Zaidi, "DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation," *Int. J. Inf. Secur.*, vol. 18, no. 6, pp. 761–785, Dec. 2019, doi: 10.1007/s10207-019-00434-1.
[23] J. Singh, S. Bagga, and R. Kaur, "Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 1970–1980, 2020, doi: 10.1016/j.procs.2020.03.226.
[24] A. Zheng, Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. Sebastopol, CA: O'Reilly Media, 2018.
[25] F. Alharbi and A. Vakanski, "Machine Learning Methods for Cancer Classification Using Gene Expression Data: A Review," *Bioengineering*, vol. 10, no. 2, p. 173, Jan. 2023, doi: 10.3390/bioengineering10020173.
[26] T. Li, G. Kou, and Y. Peng, "Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods," *Inf. Syst.*, vol. 91, p. 101494, Jul. 2020, doi: 10.1016/j.is.2020.101494.
[27] J. Zhang, P. Cheng, Z. Li, H. Wu, W. An, and J. Zhou, "A feature engineering method for machine learning inspired by quantum mechanics," in *2023 International Joint Conference on Neural Networks (IJCNN)*, Jun. 2023, pp. 1–8. doi: 10.1109/IJCNN54540.2023.10192017.
[28] J. M. Noshay *et al.*, "Quantum biological insights into CRISPR-Cas9 sgRNA efficiency from explainable-AI driven feature engineering," *Nucleic Acids Res.*, vol. 51, no. 19, pp. 10147–10161, Oct. 2023, doi: 10.1093/nar/gkad736.
[29] R. R. Shivwanshi and N. Nirala, "Quantum-enhanced hybrid feature engineering in thoracic CT image analysis for state-of-the-art nodule classification: an advanced lung cancer assessment," *Biomed. Phys. Eng. Express*, vol. 10, no. 4, p. 045005, Jul. 2024, doi: 10.1088/2057-1976/ad4360.
[30] F. M. Firnando, D. R. I. M. Setiadi, A. R. Muslikh, and Syahroni Wahyu Iriananda, "Analyzing InceptionV3 and InceptionResNetV2 with Data Augmentation for Rice Leaf Disease Classification," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 1–11, 2024.
[31] F. O. Aghware *et al.*, "Enhancing the Random Forest Model via Synthetic Minority Oversampling Technique for Credit-Card Fraud Detection," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 407–420, Mar. 2024, doi: 10.62411/jcta.10323.
[32] P. Sundarreson and S. Kumarapathirage, "SentiGEN: Synthetic Data Generator for Sentiment Analysis," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 461–477, Apr. 2024, doi: 10.62411/jcta.10480.
[33] Q. Ning, X. Zhao, and Z. Ma, "A Novel Method for Identification of Glutarylation Sites Combining Borderline-SMOTE With Tomek Links Technique in Imbalanced Data," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 19, no. 5, pp. 2632–2641, 2022, doi: 10.1109/TCBB.2021.3095482.
[34] B. V. Ramana and R. S. Kumar Boddu, "Performance Comparison of Classification Algorithms on Medical Datasets," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2019, pp. 0140–0145. doi: 10.1109/CCWC.2019.8666497.
[35] P. Kumar and R. S. Thakur, "Liver disorder detection using variable- neighbor weighted fuzzy K nearest neighbor approach," *Multimed. Tools Appl.*, vol. 80, no. 11, pp. 16515–16535, May 2021, doi: 10.1007/s11042-019-07978-3.
[36] X. Li, D. Li, Y. Deng, and J. Xing, "Intelligent mining algorithm for complex medical data based on deep learning," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 2, pp. 1667–1678, Feb. 2021, doi: 10.1007/s12652-020-02239-w.
[37] C. Li, S. Zhang, and D. Wang, "Specific Data Mining Model of Massive Health Data," in *Social Computing*, 2016, pp. 632–640. doi: 10.1007/978-981-10-2053-7_56.
[38] S. Scardapane, R. Altilio, V. Ciccarelli, A. Uncini, and M. Panella, "Privacy-Preserving Data Mining for Distributed Medical Scenarios," in *Multidisciplinary Approaches to Neural Computing*, 2018, pp. 119–128. doi: 10.1007/978-3-319-56904-8_12.