A Novel ML Approach for Computing Missing Sift, Provean, and Mutassessor Scores in Tp53 Mutation Pathogenicity Prediction

Rashmi Siddalingappa, Sekar Kanagaraj
Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India

***Abstract*—Cancer is often caused by missense mutations, where a single nucleotide substitution leads to an amino acid change and affects protein function. This study proposes a novel machine learning (ML) approach to calculate missing values in the tp53 database for three computational methods: SIFT, Provean, and Mutassessor scores. The computed values are compared with those obtained from the imputation method. Using these values, an ML classification model trained on 80,406 samples achieves an accuracy of 85%, while the impute method achieves 75%. The scores and statistics are used to classify samples into five classes: Benign, likely pathogenic, possibly pathogenic, pathogenic, and a variant of uncertain significance. Additionally, a comparative analysis is conducted on 58,444 samples, evaluating six ML techniques. The accuracy obtained by each of these is mentioned alongside the algorithm: logistic regression (89%), k-nearest neighbor (99%), decision tree (95%), random forest (99.8%), support vector machine with the polynomial kernel (91%), support vector machine with RBF kernel (84%), and deep neural networks (98.2%). These results demonstrate the effectiveness of the proposed ML approach for pathogenicity prediction.**

Keywords*—Decision tree (DT); deep neural networks (DNN); imputation; k-nearest neighbor (KNN); logistic regression (LR); missense mutations; Mutassessor; pathogenicity; Provean; random forest (RF); SIFT; support vector machine (SVM)

I. Introduction

Years of research have identified the tp53 gene, a tumor suppressor gene that encodes the tumor protein p53 (tp53), as a significant barrier in cancer development [1][2][3]. The tp53 protein acts as a tumor suppressor by regulating cell division, growth, and apoptosis processes. It has been found that approximately 90% of cancer cases exhibit tp53 mutations [4]. Notably, the mutations commonly occur between positions 102-292, resulting in approximately 190 mutated codons, with over 60% of them being missense mutations [5]. Studies by Fiamma Montovani et al. discuss the role of mutant p53 proteins in supporting malignant cell survival and cancer evolution, as well as therapeutic opportunities related to tp53 missense mutations [6]. Gaoyang Zhu et al. explore therapeutic options targeting the Gain-of-Function (GOF) feature of full-length p53 mutant proteins [7]. Additionally, Alvarado-Ortiz E et al. investigate the impact of mutp53 on metabolic reprogramming and the Warburg effect observed in cancer cells, highlighting chemo-resistance and the role of autophagy in survival [8]. Xiang Zhou et al. identify tp53 hotspots as potential barriers for novel cancer therapies and study the mechanisms underlying GOF for p53 [9]. Furthermore, cancer cells employ various strategies to disarm p53 and promote their growth and survival [10]. One approach involves mutating the tp53 gene itself, removing the protective function and allowing unmonitored cell activities [11]. Nonsynonymous Single-Nucleotide Variants (nsSNVs) are considered a primary reason for cancer, as they alter proteins with a single residue change in the amino acids [12][13]. Yong Li et al. demonstrate the predictive value of tp53 in the untranslated region (UTR) of cancer specimens, highlighting the impact of germline SNVs on tp53 protein levels and cell apoptosis [14]. Oliver Poirion et al. propose using expressed SNVs (eSNVs) from RNA sequences to locate tp53 variations in tumor subpopulations [15]. Computational procedures have been developed to assess the influence of amino acid substitutions and the frequent occurrence of missense variants in cancer patients [16] [17]. Understanding the effect of missense mutations is crucial for clinical use, especially in distinguishing pathogenic and infectious variants among numerous missense variants.

II. Related Work

With the rapid development of Machine Learning (ML) and its applications in various fields, ML has emerged as a potential solution for cancer research [19][20]. Efforts have been made to apply ML/AI-based diagnostics for cancer using vast genomic data. Techniques such as REVEL, CADD, FATHMM, and PolyPhen employ ML algorithms like Random Forest (RF), Naïve Bayes (NB), and Logistic Regression (LR) to predict pathogenicity [21][22]. Jiaying Lai et al. introduce LYRUS, a machine-learning tool that predicts pathogenicity based on missense variants [23]. LYRUS utilizes an XGBoost classifier incorporating sequence, structure, and dynamic features. The tool is evaluated using F-scores and specificity metrics, outperforming alternative methods. However, LYRUS estimates pathogenicity based on the actual protein structure and does not consider the mutated protein. It is also limited to proteins with available structures in the Protein Data Bank (PDB). Hua Tan et al. differentiate cancer-causing driver mutations from normal ones using SVM classification based on distinguishing features [24]. Their approach demonstrates higher efficiency compared to existing methods. In clinical research, computational techniques such as SIFT, Mutassessor, and Provean are used to predict the pathogenicity of missense mutations. However, there is a lack of ML-based methods to calculate these scores. Therefore, the present study proposes a novel approach to calculate SIFT,

Provean, and Mutassessor scores using K-nearest neighbors (KNN) regression. The study also focuses on classifying samples into pathogenicity classes based on the guidelines suggested by the American College of Medical Genetics and Genomics (ACMG) [25]. Section III of the paper delves into the materials and methods utilized in the research study. Following this, Section IV elaborates on the implementation of the algorithms employed. The subsequent section, Section V, presents the results and output obtained from the study, providing a detailed analysis. Finally, in Section VI, the paper concludes by summarizing the main findings and implications, offering a comprehensive conclusion to the research.

III. Materials and Methods

A. Computational Techniques for Pathogenicity Prediction

1) SIFT score: The SIFT (Sorting Intolerant from Tolerant) method is a prediction tool that assesses the relationship between amino acid substitutions and protein functions [26]. It is based on the hypothesis that amino acids tend to be conserved within a protein family. Therefore, any changes at well-conserved amino acid positions are likely to be damaging. SIFT also considers the presence of hydrophilic amino acids, such as valine, and checks if the substituted amino acid is another hydrophilic amino acid, like isoleucine or leucine. In such cases, the changes are predicted as tolerated. However, substitutions to other types of amino acids are assumed to result in functional changes. The SIFT method takes the protein sequence as input and aligns it with related proteins. It calculates the probability of amino acid occurrence at each position during the alignment. If the probability falls below a certain threshold, SIFT predicts the substitution as deleterious, otherwise, it is considered tolerated. The threshold value typically ranges from 0.0 to 1.0, where scores between 0.0 and 0.05 are considered deleterious, and scores greater than 0.05 are considered tolerated.

2) Provean score: The Provean (Protein Variation Effect Analyzer) score operates similarly to the SIFT method [27]. It calculates an alignment score for each protein sequence. A set of closely related sequences, typically the top 30 clusters, is selected as a supporting sequence set. The scores within each cluster are averaged, resulting in a Provean score. This score is then compared to a predefined threshold, typically set as -2.5. If the score is equal to or lower than the threshold, the protein variant is considered deleterious; otherwise, it is considered "neutral."

3) Mutassessor score: The Mutassessor score (Mutation Accessor) predicts the functional impact of an amino acid change based on the evolutionary conservation of the affected amino acid among protein homologs [28]. The default threshold for pathogenicity classification is set to -1.93, distinguishing high or medium functional impact variants from low or neutral predicted variants.

Note: These scores, namely SIFT, Provean, and Mutassessor, are utilized in computational techniques to predict the pathogenicity or functional impact of missense mutations in proteins.

B. The Proposed ML-based Method to Calculate the Missing Values of SIFT, Provean, and Mutassessor Scores

In this section, two algorithms related to the present research study are discussed. Algorithm-1 presents the proposed ML-based approach for calculating missing values of three different computational scores. Algorithm-2 outlines the process of classifying each sample into pathogenicity classes. The classification results are compared using six different ML techniques.

Algorithm – 1: Proposed algorithm for predicting the missing values of Sift, Provean, and Mutassessor Scores in tp53 database

Input: tp53 mutation samples (80346, 133) → 80346 rows X 133 columns; Output: Predicted scores for the missing values in Sift, Provean, and Mutassessor scores

Step 1: Preprocess the tp53 original dataset.
Step 2: Perform feature selection to select the features required for the proposed task.
Step 3: Separate rows with and without Sift scores.
Step 4: Consider the rows that have Sift scores.
i. Create a dataframe (x_train) to store the features.
ii. Create another dataframe (y_train) to store the corresponding labels.
iii. Use the KNN regressor model to predict values of y_train, and save the predictions as y_predict.
iv. Compute the Mean Absolute Error (MAE) score of y_train and y_predict for each 'k' value from 2 to 20.
v. Determine the 'k' value with the minimum MAE score among all the MAE scores.
vi. Train a new model using this 'k' value and save it as final_model.
Step 5: Use final_model to calculate the missing values of Sift scores from step 3 using the KNN regressor technique:
i. Consider the complete feature set of missing and present Sift values.
ii. Calculate the Euclidean distance (ED) for each feature set where Sift scores are present and where Sift scores are missing.
iii. Tabulate all ED values in ascending order.
iv. Select the top 'k' values (from step 4.vi).
v. Calculate the average of these scores and save it as the new predicted Sift score.
vi. Return the new predicted Sift score.
Step 6: Predict Sift scores using all the features selected in step 2 with the help of the impute method.
Step 7: Compare the final predicted values from steps 5 and 6.
Step 8: Repeat steps 3-5 to determine Provean scores.
Step 9: Repeat steps 3-5 to determine Mutassessor scores.
Step 10: Stop .

Algorithm – 2: Classification of samples into five classes of pathogenicity using different ML techniques

Input: tp53 mutation samples.
Output: Pathogenicity classification.

Step 1: Choose features and labels from the tp53 database (features computational scores + stat scores).
Step 2: Remove samples with null values.
Step 3: Perform the classification of each sample into pathogenicity classes using the following ML techniques:
i) Logistic regression,
ii) KNN,
iii) SVM,
iv) Decision tree,
v) Random forest,
vi) Feedforward neural network.
Step 4: Compare the results of each technique using evaluation metrics.
Step 5: Tabulate the results.
Step 6: Stop.

C. ML Techniques used in the Proposed Research Study

- To predict the computational scores

1) K-Neighbors Regressor: This technique is a regression method derived from the KNN model. It calculates values based on the representation of the 'k' nearest neighboring target values from the training dataset. The values present in the training class are stored, while those that are missing are later calculated using similarity scores such as Euclidean, Manhattan, or Hamming distance. The accuracy of the calculated values relies on the selection of a primary measure, 'k'. Choosing an appropriate 'k' value is crucial, as a large 'k' value can potentially exploit the distance boundaries and result in overfitting or blurring of the feature space. Conversely, a low 'k' value can lead to underfitting of the model [29]. Hence, an optimal 'k' value is determined by discarding the missing values from the target variable field and predicting the target variable values using different 'k' values. These predicted values are then compared with the actual target values, and the difference is evaluated using the Mean Absolute Error (MAE) score. The 'k' value that yields the lowest MAE score is selected as the final 'k' value for the K-Neighbors Regressor. Table I provides a tabular representation of the procedure.

TABLE I. THE KNN REGRESSOR METHOD WAS USED TO CALCULATE THE MISSING VALUES. THE TABLE SHOWS THE SAMPLE VALUES TAKEN FROM THE TP53 DATABASE. IT CONTAINS A COMBINATION OF VALUES PRESENT AND ABSENT INDICATED WITH DIFFERENT COLORS

data_pre➔train_X, train_y, data_abs➔xdim ,ydim,

L_stat	C_stat	T_stat	G_stat	S_stat	Sm_stat	Sift_score	ED
0.01	0.08	0.05	0.44	0.71	0.331	0.19	0.34
2.84	2.80	2.87	2.77	1.40	2.107	0	5.83
0	0.00	0.00	0.91	0	0.01	?	0.34
0.02	0.03	0.03	0	0.03	0.083	0.89	0.9155

Note: L: Leukaemia, C: Cell_line, T: Tumor, G: Germline, S: Solid_state, Sm: Somatic, ED: Euclidean Distance

Calculating ED individually for rows (i), (ii), and (iv) containing SIFT score values and SIFT score=? Different arrows indicate this in Table I. Below is the ED calculation for row (i).

$$\sqrt{(0.014-0)^2+(0.082-0.001)^2+(0.053-0.001)^2+(0.071-0)^2+(0.331-0.01)^2} = 0.342$$

Likewise, EDs for all the rows (ii and iv) w.r.t data_pre

Sort ED: 0.34, 0.91, 5.83. Consider, k=2, so pick the first 2 points and take the average.

$$\frac{0.34+0.91}{2} = 0.625$$

The new sift_score predicted is 0.625

D. To Classify Samples into Various Pathogenicity Classes

- *Feature selection:* With the help of data visualization and pre-processing using principal component analysis (PCA), the dataset was prepared for the training phase [30]. With PCA, highly correlated features (both positive and negative) were removed from the original dataset. For the strongly correlated features, only one of the features is retained. To decide this, the following aspects were identified; if two features are to -1, they are negatively correlated, and if the values are closer to +1, they are positively correlated. After performing the feature reduction process, the dataset had 58444 X 10 records that were finally used for the classification process using six different ML techniques. In the end, each ML technique is compared to study the best method for classifying a sample. The model was evaluated using F-score and parameter tuning to ensure robustness. Finally, the models are evaluated on the test set for full and reduced features. Feature reduction, indeed, has an impact on the overall algorithm performance of these ML techniques. Fig. 1 depicts the framework of this modeling process. The implications of these methods are described below.

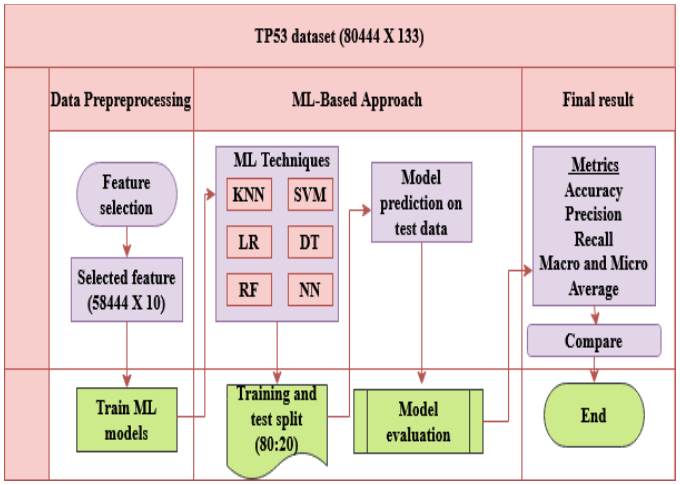


Fig. 1. The proposed schematic hybrid framework of the modelling process to predict the pathogenicity of a sample using tp53 database and various ML algorithms such as Logistic Regression (LR), K-nearest neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF) and lastly, Feed-Forward Neural Network (NN).

- *Logistic regression:* The LR is, by default, a regression model whose prediction is based on the logistic function [31]. The decision is associated with the probability that a given feature belongs to some categorical class, say, 1. If a sigmoid logistic function is used to make the prediction, then if sigmoid function (S) resorts to an infinite value when the prediction variable (ŷ) will become one and ŷ will be 0 if 'S' is a negative value, given by Eq. 1

$$sig(s) = \frac{1}{1+e^{-x}} \quad (1)$$

A crucial parameter in logistic regression for the present classification task is multinomial data distribution since the categories of the classes (5 pathogenicity classes) are without any specific ordering. The classification of a sample is performed based on the threshold. The threshold value is vital in estimating the probability that a sample belongs to one out of these five classes. Say if ŷ ranges between 0-0.2, then the sample is classified as '0 - benign', for ŷ between 0.2 – 0.4, the sample will be classified as '1-LP', with a range between 0.4 - 0.6 the class will be '2-P', 0.6 – 0.8 for class '3-PP' and finally 0.8 – 1.0 for class '4-VUS'. This is usually the first ML algorithm to be used for any classification task.

- K-Nearest Neighbors: This is the simplest of all the ML techniques that intend to classify a record (unlabelled) based on the class of the neighbouring data points (labelled) [32]. Using a distance measure, say ED, the distance between the features of the unlabelled and labelled records is calculated. Using an optimal 'k' value, the nearest top 'k' neighbours are chosen. Finally, the class label with the highest number is tagged for the unlabelled data point. The main idea behind this intuition is that similar points tend to be close to each other. As this is a multi-class classification problem, a sample will be classified into one of the five classes. The best 'k' value obtained on the dataset is 5. Thus, k=5 was used to train the final model.

- Support Vector Machine (SVM): SVM is a versatile algorithm used for classification and regression tasks. It aims to find an optimal hyperplane, or decision boundary, that maximizes the separation between different classes [33]. When classes are not linearly separable, SVM employs the kernel trick, using functions like linear, polynomial, RBF, or sigmoid. Data points close to the hyperplane are called support vectors. For multi-class classification, SVM utilizes the one-vs.-one approach, explicitly indicated by defined_function_shape=ovo. By default, it uses the one-vs.-rest approach (defined_function_shape=ovr), where data points of one class are compared with the rest [33]. In our case with five pathogenicity classes, SVM is applied using both 'rbf' and 'poly' kernels, with specified parameters such as gamma=0.5, C=0.1, and degree=3 for 'rbf', and C=1 for 'poly'.

- Decision Tree: This rule-based classifier resembles a tree-like structure and makes decisions based on a series of questions. At each node, a question is asked, and depending on the answer (yes or no), the algorithm progresses to other nodes at subsequent levels, similar to an if-else structure. Decision trees consider one feature at a time from the input data (X) to create branches. The feature can be categorical or continuous, using categories or thresholds as decision criteria. Different criteria, such as Gini impurity and Entropy, can be used to determine the root node and subsequent decision-makers. Gini impurity calculates the frequency at which a sample in the dataset will be incorrectly labeled, while Entropy measures the disorder of features (X) with respect to the target label (y) [34].

$$Gini\ Impurity = 1 - \sum_i p_i^2 \quad (2)$$

$$Entropy = -\sum_i p_i og_2\, p_i \quad (3)$$

Where Pi is the probability for class 'i' such that i=1 to 5. In the present study, the question would be: 'is the leukemia_stat greater than a threshold value, say, x? Or is leukemia_stat less than or equal to the threshold value? Thus, the DT will traverse each node and evaluate the condition before deciding which branch to proceed with until the leaf node (last) is hit. Here, there will be a total of five leaf nodes for each pathogenicity class. Both entropy and Gini impurities are used separately in the present study with max_depth=3.

- Random Forest: It is based on the concept of ensemble algorithms, which combines multiple classifiers, and decision trees, solves the problem independently, and combines the results in the last step [35]. With this approach, the overall performance is improved. The model with correct prediction is retained, and incorrect predictions are pruned. The prediction rules are not visible to the user, thus enforcing a black-box concept. The multiple final DTs are combined, and the class with a majority vote will be assigned to the sample. With multiple DTs, the model obtains a higher accuracy and eliminates the problem of overfitting. RF will achieve the best accuracy compared to the previous models discussed here. The following parameters are used in the present study; n_estimators=100 (overall trees the forest has), bootstrap = True (randomize the samples in the dataset), max_features = 'sqrt' (takes the square root of the total features present in the dataset. Total features = 10 (computational scores+stat values + pathogenic class). $\sqrt{10} \sim 3$, so three features are tried randomly for each tree).

- Artificial Neural Network: ANN represents the working of a real human brain where the brain will generate outputs based on the past information trained earlier in life. ANN is suitable for any function, especially datasets that exhibit non-linear relationships. Feedforward neural network is a variation of ANN with three layers, an input layer, one or more hidden layers, and an output layer. Every layer has multiple nodes/neurons to process the input. The neural networks learn when fed with input and propagate to subsequent layers; hidden and output. This is called the learning/training phase. At each node at every layer, the network calculates the product of input values and weights, and the sum of these product terms along with

a bias value is calculated at every hidden node and sends the value to the next layer. That is, the network calculates a function, say 'f', for a predetermined input feature in 'X' and results in a training pair (X,y) such that f(X) ≈ (y). The actual and predicted values are calculated to understand the loss incurred by the network [36]. At the output layer, an activation function is used to obtain the result. The activation functions are: Sigmoid (the output value ranges between 0 and 1), tanh (ranges between -1 and +1), Rectified Linear Unit (ReLU) (returns the max (0, X)), softmax (return the probability of belonging to each output class, such that, when the values are added, we get 1). In the present study, a simple sequential model is trained using Keras that uses TensorFlow objects. The input_dim was set to 9, matching the number of input parameters (computational scores + stat values), and the activation was ReLU with 16 neurons in the input layer. Two hidden layers were used, each with 32 and 64 neurons and the same activation function. The output layer has five neurons as there were five pathogenicity classes with softmax activation. The loss function was "sparse_categorical_crossentropy", optimizer='adam', metrics were set to accuracy with 100 epochs.

IV. Implementation

A. Dataset Collection

The dataset used in this study was collected from the UMD-tp53 database (Universal Mutation Database). The database, which initially had only 360 mutations in 1992, has now grown to contain over 80,000 mutation samples [37]. It consists of two files: variant and mutation. The mutation database includes samples of all patients with a tp53 mutation, while the variant database contains unique tp53 variants found in these patients. For this study, the mutation database with 80,406 samples (TP53 Mutated data, 2017 Release R2, available at https://p53.fr/the-database) was utilized. The database includes various variant classifications for mutant types, such as missense (58,517), nonsense (8,460), Frame-shift-del (5,212), splice-site (2,348), synonymous (2,016), frame-shift-ins (1,701), Indel (1,194), Ins (290), and others (668). The database was downloaded in CSV format.

B. Data Pre-Processing Phase

The initial mutant database downloaded from the tp53 website consisted of 80,406 rows and 133 columns. The prediction scores were based on various statistical values and computational scores present in the database. However, when the features start_DNA and end_DNA had a value of '?', most of the remaining features also had '?' (119 columns), and the pathogenicity class was labelled as 'no prediction.' Therefore, the rows with values start_DNA and end_DNA = '?' were removed as the first step in the pre-processing phase. This resulted in 80,346 rows and ten columns. Additionally, the start and end_DNA features were not used in the prediction or classification process, so they were dropped from the feature set, resulting in a final dataset size of 80,346 X 8. The next step in pre-processing was to handle null values. Although there were no null values, three features (Sift, Mutassessor, and Provean scores) contained string values such as 'No data,' 'No protein,' 'Not known,' and 'Inframe.' As part of data cleaning, these string values were replaced with '?', as these values would be calculated using the proposed algorithm. Furthermore, the pathogenicity feature consisted of categorical data such as benign, likely pathogenic, pathogenic, possibly pathogenic, and VUS. To handle this, a label encoder was used to transform the string values into integer values. The respective classes were assigned the numbers 0, 1, 2, 3, and 4.

C. Data-Splitting:

The new DataFrame (new_df) with a size of 80,346 X 8 was further divided into two DataFrames: data_abs, which contained rows where the Sift_score was '?', with a size of 21,902 X 8, and data_pre, which included rows with available Sift_score values, with a size of 58,444 X 8. From data_pre, the features and labels were separated and named data_pre_temp and 'y', respectively. The '.values' function was used to convert the DataFrame data_pre_temp into a list named Xin. The KNeighborsRegressor class was then employed to train the model using Xin as the input features and y as the target labels in an 80:20 ratio. To find an ideal 'k' value, the 'k' value was varied from 2 to 20, and the Mean Absolute Error (MAE) was calculated for each 'k' value. The MAE represents the mean absolute difference between the actual and predicted values. The 'k' value that yielded the lowest MAE value was considered the optimal 'k' value for training the final model to predict the missing values. The DataFrame data_abs was split into data_abs_temp (features) and ydim (labels). The '.values' of data_abs_temp were stored in Xdim as features, with ydim representing the labels. A new DataFrame named data_predict was created with a column of the same name, Sift-score, to store the predicted values of ydim. This DataFrame was then joined with data_abs_temp and renamed as 'dataframe_1'. The values of Sift_score were extracted from data_pre and stored in a new DataFrame called df_join, which was further joined with data_pre_temp and renamed as 'dataframe_2'. Finally, dataframe_1 and dataframe_2 were concatenated to form a new DataFrame named 'dataframe' with a size of 80,346 X 8, which matched the original size of the initial DataFrame new_df. The DataFrame 'dataframe' now contained values that originally had missing values (21,902)

V. Results

The predicted values obtained using the proposed algorithmic approach were compared with the state-of-the-art ML library method called Impute. KNNImputer was utilized with the same 'k' value as in the previous method. The values calculated by both methods were compared, and it was found that they were 85% similar. Additionally, two KNN models were trained separately, one using the proposed method and the other using the imputer method. The proposed model demonstrated superior accuracy compared to the built-in method.

A. Evaluation of Computational Scores Prediction using the Proposed Method and Built-In Method

The objective is to develop an ML-based approach to calculate missing values in three important pathogenicity prediction methods based on amino acid substitutions in protein sequences. In the tp53 database, certain values for

these three features were missing. Instead of using existing algorithms, this study employs the KNN regressor, an ML technique, for estimating these values. Additionally, each method requires a threshold, which can be adjusted based on user requirements. Hence, the threshold value was redefined to align with the existing value range. Table II presents the threshold used in this study to classify the scores into their respective variant classes. Fig. 2(a) to 2(c) shows the graphical illustration of the values computed for all three computational scores from both methods impute and code-based.

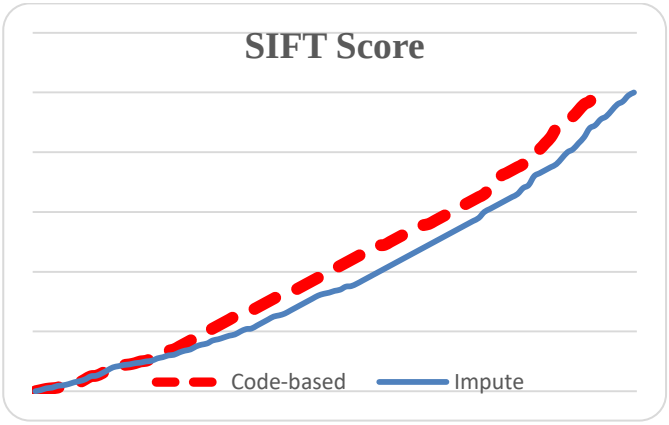
TABLE II. THE THRESHOLD VALUES ARE USED FOR DIFFERENT COMPUTATIONAL METHODS IN THE PATHOGENICITY CLASSIFICATION TASK

Computational Methods	Threshold values: Class type		
Sift	<=0.05: **Harmful**	>0.05: **Tolerated**	--
Provean	<=2.5: **Deleterious**	>2.5: **Neutral**	--
Mutassessor	<=1.0: **Neutral**	>1.0 &<=2.0: **Low**	>2.0 &<=4.0: **Medium**

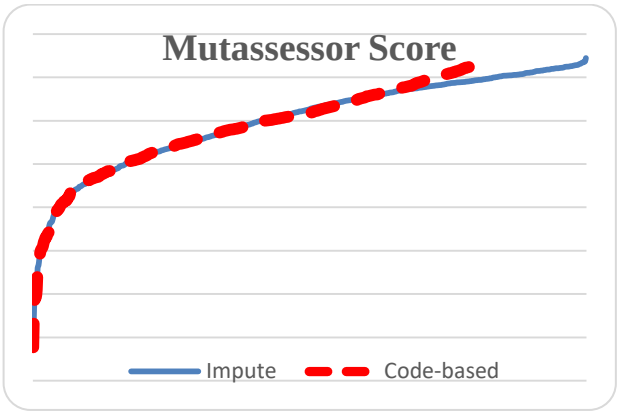
Note: Shown in bold letters are the category labels used for each of the threshold values

Do the values computed by the proposed procedure outperform the reference method? - A Case study:

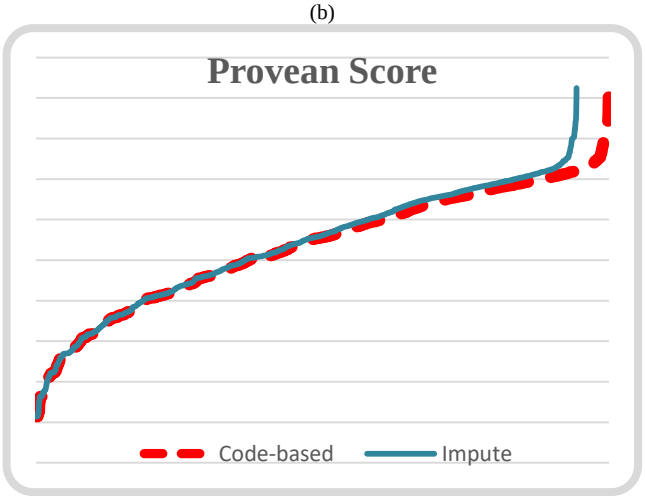
As depicted in Fig. 2, the computed missing values from both methods closely align, with minor variations observed at the beginning and end of the graph. However, the question arises whether these slight differences hold any predictive significance. Therefore, a case study was conducted to demonstrate that the proposed method exhibits superior classification performance for tp53 mutation samples. After calculating the missing values, an SVC classifier was employed to classify the samples based on pathogenicity variants using the computational methods. To further assess the results, the impute method, an ML library method for calculating missing values, was employed, and the same process was repeated. The trained SVC classifier effectively classified the samples using both the code-based and impute methods. The code-based approach achieved higher classification accuracy compared to the existing impute method for all three computational techniques. Additionally, the match percentage for each variant class was also calculated. The proposed and built-in methods achieved a match rate of over 81%. The significance of this evaluation is summarized in Table III.



(a)



(b)



(c)

Fig. 2. (a) SIFT scores computed using code-based and reference methods (impute)., (b) Mutassessor scores computed using code-based and reference methods (impute)., (c) Provean scores computed using code-based and reference methods (impute).

TABLE III. THE NUMBER OF SAMPLES CLASSIFIED TO EACH PATHOGENICITY LABEL FOR BOTH PROPOSED AND BUILT-IN METHODS. THE CLASSIFICATION ACCURACY IS THE MEASURE CALCULATED FOR THE CLASSIFIED DATA IN COLUMN-WISE, REPRESENTED IN BLUE COLOUR. THE GREY COLOUR FIELD REPRESENTS THE PERCENTAGE OF A MATCH IN THE VALUES CALCULATED BY BOTH APPROACHES

Computational Method		ML-based proposed approach	Built-in impute method	% of a match between proposed and built-in method
Sift	Damaging	74761	73092	85.32
	Tolerated	5585	7254	
	Classification Accuracy	0.879	0.764	
Provean	Deleterious	72733	71838	81.91
	Neutral	7613	8508	
	Classification Accuracy	0.875	0.781	
Mutassessor	Medium	73810	73894	84.89
	Low	4539	4203	
	Neutral	1997	2249	
	Classification Accuracy	0.872	0.783	

B. Evaluation Metrics to Assess ML Model Performances

TABLE IV. THE NUMBER OF SAMPLES IN EACH PATHOGENICITY CLASS FOR THE TRAINING AND TEST DATASET

0:BENIGN, 1:LIKELY PATHOGENIC, 2: PATHOGENIC, 3: POSSIBLY PATHOGENIC, 4:VUS

	Data split 80:20	Class #				
		0	1	2	3	4
No. of training samples	46755 80%	50	5146	30509	7981	3069
No. of test samples	11689 20%	11	1303	7636	1998	741
Total	58444 100%	61	6449	38145	9979	3810

Table IV gives the number of samples in each pathogenicity class for the training and test dataset.

Confusion Matrix (CM) is a tabular representation of the performance in the classification task [38]. It contains the true values along the y-axis and estimated values along the x-axis. The number of rows and columns depends on the number of classification classes.

TABLE V. A CONFUSION MATRIX FOR A RANDOM FOREST ALGORITHM FOR MULTI-CLASS CLASSIFICATION OF PATHOGENICITY LABELS

N REPRESENTS A CLASS NAME; CM IS THE CONFUSION MATRIX C. A GREEN COLOUR ROW REPRESENTS AN FN, AND THE YELLOW COLUMN REPRESENTS AN FP, AND PINK IS THE ACTUAL TRUE POSITIVE FOR THE CLASS N=1. ACTUAL CLASS : AC

CM(C)		**Prediction Class**					
	N classes	N=0	N=1	N=2	N=3	N=4	Total
AC	N=0	CC_{00}=11	0	0	0	0	11
	N=1	0	CC_{11}=1293	0	0	10	$A_{N=2}$=1303
	N=2	0	0	CC_{22}=7636	0	0	7636
	N=3	0	0	0	CC_{33}=1994	4	1998
	N=4	0	4	0	0	CC_{44}=737	741
	Total	11	$P_{N=2}$=1297	7636	1994	751	T=11689

Table V describes a CM matrix of the RF algorithm, illustrating the different numbers obtained from the ML model. Here, CCNN indicates the correctly classified samples, T is the count of test samples, AN is the total times a sample is correctly classified to its actual class, and PN represents the number of times a sample is predicted. The main components of a CM are as follows: A true positive (TP) is when a true class 0 (benign) is predicted as 0 (benign). A true negative (TN) is when an actual class is not 0 and is predicted correctly as not class 0. A false positive (FP) is when a true class 0 is wrongly predicted as class 1 or any other class, and lastly, a false negative (FN) is when a true class is not 0 but is mispredicted as class 0. Further, the standard performance metrics derived from CM are described in Eq. [4 – 7]. Those are i) A recall is a measure of all positive samples that the model predicted correctly for the class; this indicates how much the model correctly predicted for the total samples of class 0. ii) A precision indicates the quality of the prediction, i.e., how many times the model correctly predicted a sample as class 0 out of all the total number of class 0 true samples. iii) F-Score is the average of both recall and precision. iv) accuracy is the actual number of samples that the model correctly classifies over the total number. v) The macro average scores are calculated by considering the weighted mean for each R, P, and F for every predicted class without considering each label's proportion. vi) The weighted average score is calculated by taking the product of the sum of individual recall, precision, and f-score and each classified sample over the actual number of samples for the classification class. This is similar to the macro score except that the weighted score considers the proportion of individual labels. vii) The micro average considers the total TP, FP, and FN irrespective of the prediction made by the model for each class

$$Recall\ (R) = \frac{TP}{TP+FN} \quad (4)$$

$$Precision\ (P) = \frac{TP}{TP+FP} \quad (5)$$

$$F\ Score = 2 * \frac{PR}{P+R} \quad (6)$$

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (7)$$

Table VI illustrates the performance achieved for each of the ML techniques on the test dataset.

TABLE VI. THE TABULATION OF VARIOUS EVALUATION METRICS ON THE TEST DATASET FOR EACH ML METHOD. THE RF RESULTED IN THE HIGHEST ACCURACY, CLOSELY FOLLOWED BY KNN AND DL METHODS

Method	Class	P	R	F		Macro	Micro	Weighted	Accuracy
KNN	0	1.00	1.00	1.00	P	0.99	0.99	0.99	0.994
	1	0.99	0.99	0.98	R	0.98	0.99	0.99	
	2	1.00	1.00	1.00	F	0.98	0.99	0.99	
	3	0.97	1.00	0.99					
	4	0.96	0.95	0.95					
SVM	0	1.00	1.00	1.00	P	0.86	0.91	0.91	Poly: 0.910 RBF: 0.84
	1	0.78	0.74	0.76	R	0.84	0.91	0.91	
	2	0.99	0.99	0.99	F	0.85	0.91	0.91	
	3	0.74	0.85	0.79					
	4	0.80	0.61	0.70					
LR	0	1.00	1.00	1.00	P	0.85	0.89	0.89	0.891
	1	0.84	0.48	0.61	R	0.79	0.89	0.89	
	2	0.99	0.99	0.99	F	0.81	0.89	0.89	

	3	0.67	0.89	0.76					
	4	0.77	0.60	0.67					
DT	0	1.00	1.00	1.00	P	0.72	0.95	0.95	Gini:0.954 Entropy:0.952
	1	0.92	0.87	0.90	R	0.72	0.95	0.95	
	2	1.00	1.00	1.00	F	0.72	0.95	0.95	
	3	0.88	0.90	0.89					
	4	0.78	0.81	0.80					
RF	0	1.00	1.00	1.00	P	0.99	1.00	1.00	0.998
	1	0.99	0.99	0.98	R	1.00	1.00	1.00	
	2	1.00	1.00	1.00	F	1.00	1.00	1.00	
	3	1.00	1.00	1.00					
	4	0.98	0.99	0.99					
DL	0	1.00	1.00	1.00	P	0.96	0.98	0.98	0.982
	1	0.97	0.96	0.96	R	0.97	0.98	0.98	
	2	1.00	1.00	1.00	F	0.96	0.98	0.98	
	3	0.97	0.95	0.96					
	4	0.88	0.92	0.90					

Cross-validation is the most famous evaluation metric to estimate the actual prediction of an ML model [39]. This method splits the entire dataset into ten folds (k-cross fold where k=10) to form a training and test set with 0-9 folds consisting of 0 - 5844 samples and the 10th fold containing 5845 - 5848 samples. After executing the final model 10 times, all ten folds accuracy scores were obtained using cross_val_score (Table VII). The average scores for all 10-folds are obtained using cross_val_predict.

TABLE VII. TABULATION OF ACCURACY FOR EACH ML METHOD FOR EACH FOLD IN CROSS-VALIDATION APPROACH. THE K VALUE IS 10, WHERE 0-9 FOLDS RANDOMLY SERVE AS THE TRAINING SET, AND THE REMAINING ONE FOLD ACTS AS A TEST SET

	1	2	3	4	5	6	7	8	9	10
KNN	0.93	0.92	0.92	0.88	0.92	0.94	0.92	0.91	0.88	0.94
LR	0.82	0.80	0.94	0.86	0.81	0.96	0.89	0.90	0.83	0.89
SVM	0.91	0.89	0.91	0.91	0.90	0.92	0.90	0.91	0.90	0.91
DT	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.96
RF	0.98	0.99	0.99	0.99	0.99	0.98	0.99	0.98	0.99	0.98
DL	0.98	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98

C. Discussions

So far, the pathogenicity of cancer types has been studied using computational scores calculated using various statistical approaches. However, the rapid growth of machine learning applications has sparked interest in designing an ML-based strategy for calculating these scores. In the first approach of this research study, three computational scores were calculated based on the data available in the tp53 database. The thresholds for these scores were kept unchanged, consistent with those used in the tp53 repository. The results were compared with the existing ML library's impute method. Subsequently, a separate KNN model was trained using the calculated scores from the code and the built-in approaches. It was observed that the code approach outperformed the built-in method in terms of accuracy. This process was repeated for all three computational techniques used to calculate the scores. Furthermore, when three or more statistical scores were equal to zero, the predicted Sift score was always zero. However, when these values were utilized for the classification task, the model achieved only 78% accuracy. Consequently, input features with a high number of zero values were dropped, and the remaining samples were considered for the classification task. In the second part of the study, six different ML techniques were evaluated to classify tp53 samples into pathogenicity classes. The investigation revealed that ML algorithms efficiently classified the data with very high accuracy in most models. Among the six algorithms, the RF algorithm yielded the best results, achieving an F-score of 1 in many cases. As mentioned in the introduction, missense mutations are highly prevalent in approximately 80% of cancer samples. Scientists worldwide dedicate their valuable time to understanding the significance of these mutations and devising novel techniques to combat cancer. Therefore, the present research study offers practical solutions in significantly less time compared to manual evaluation. In terms of clinical significance, clinicians can utilize these techniques to swiftly obtain computational scores and classify records into pathogenicity classes without the need for clinical tools or equipment intervention. Moreover, RF and NN techniques could be adopted for risk analysis and the design of predictive diagnostic procedures. Although this hypothesis was not proven in the present study, literature reports suggest that NN techniques could outperform other ML algorithms in such tasks.

1) Drawbacks: The present study has several limitations. Firstly, the proposed prediction strategy heavily relies on the existing dataset values. It can only predict missing values in a feature column, assuming that the column already contains some pre-processed values. Consequently, the predictive ability of ML models is contingent upon the values present in the database, which may result in sampling errors when applying feature selection techniques. Furthermore, the study compares the classification accuracy of six prominent ML algorithms. However, without any specific reason, other efficient ML models were not investigated. For instance, deep neural network-based models could have potentially addressed the problem of feature selection in a more effective manner. The omission of such efficient algorithms limits the comprehensive exploration of potential solutions for feature selection. These limitations should be taken into consideration when interpreting the results and implications of the study. Future research should aim to overcome these drawbacks and explore the application of additional ML models to improve

feature selection and enhance the predictive performance of the proposed approach.

2) Future work: There are several potential areas for further extension in this research study. First, it involves locating the actual disease-causing missense variants among all gene-specific mutations in a patient's sample. Typically, a single cancer patient may have approximately 500 missense mutations. However, only a few of these mutations exhibit cancer-related symptoms, while the majority may be non-cancerous or benign. ML-based models can assist in narrowing down the candidate mutations based on predictive scores, thereby reducing the time required for pathogenicity prediction and minimizing diagnostic costs. Second, a prediction model can be developed for pathogenicity classification based on different types of mutations, such as missense and frameshift mutations. Such a model can utilize amino acid sequences as input features and forecast the functional domains of genes and proteins involved in causing these deleterious mutations. Third, the focus could be on identifying the pathogenic components within a gene and searching for symptoms associated with similar diseases. This knowledge can aid in determining appropriate treatment approaches, potentially using similar strategies employed for identical diseases. It may also facilitate the process of target identification for prospective drug development. Fourth, it is important to identify the proteins involved in each malignant mutation, analyze their characteristics, and identify drugs that target these proteins in both Gain-of-function and Loss-of-function situations. For instance, in the case of tp53, Loss-of-function is considered. Fifth, incorporating patient-specific gene information can help assess interactions between genomic variants. This approach could provide a likelihood ratio for disease-causing genes and enable the targeting of these genes for effective drug interventions, further supported by in-vitro methodologies. Lastly, creating a multi-layer neural network model can enhance understanding of clinical carcinogenesis and evolutionary conservation by analyzing amino acids conserved throughout the progression. The gene and protein information obtained from previous steps can be leveraged for this prediction task.

VI. Conclusion

The present research study focused on two key aspects: estimating the missing scores using a novel ML method and comparing and analyzing different ML algorithms for a classification task. The proposed ML-based approach for calculating missing values in three pathogenicity prediction computational scores has two strong points for medical use. First, there haven't been any such algorithms to calculate these scores using an ML technique that exhibits high accuracy compared to the built-in ML library method. The other point is leveraging this idea to classify the samples from the tp53 database into their appropriate pathogenicity class, as defined by ACMG guidelines. Furthermore, missing values in databases are a common hindrance to achieving high accuracy. Thus, the proposed technique could calculate these missing values in a diverse range of databases. Additionally, the research used six different ML techniques to classify the tp53 database based on the pathogenicity class. It was found that RF and DL outperformed other methods in terms of various performance metrics. The study also suggested that logistic regression performed poorly with an accuracy of 89% compared to other techniques. The features used in this study could help unravel effective biomarkers related to the tp53 database. Clinicians may perform complementary analyses in terms of validation and clinical trials by adopting the proposed framework. The best-performing model could further be enhanced by training it on a different dataset. Once approved by standard authorities, the ML-based clinical tool may collect blood samples from patients, predict the values of computational scores, and provide the likelihood of pathogenicity. Overall, this research study offers promising insights into addressing missing values and improving classification accuracy in the field of pathogenicity prediction. The proposed ML-based approach has the potential to enhance diagnostic capabilities and facilitate personalized treatment decisions in clinical settings.

Conflict of Interest

The author(s) declare that there are no conflicts of interest for the present study.

References

[1] Blackadar C. B. (2016). Historical review of the causes of cancer. World journal of clinical oncology, 7(1), 54–86. https://doi.org/10.5306/wjco.v7.i1.54.

[2] Pineros, M., Mery, L., Soerjomataram, I., Bray, F., &Steliarova-Foucher, E. (2021). Scaling up the surveillance of childhood cancer: A global roadmap. Journal of the National Cancer Institute, 113(1). https://doi.org/10.1093/JNCI/DJAA069.

[3] Ferlay J, Ervik M, Lam F, Colombet M, Mery L, Piñeros M, et al. Global Cancer Observatory: Cancer Today. Lyon: International Agency for Research on Cancer; 2020 (https://gco.iarc.fr/today, accessed February 2021).

[4] Monti, P., Menichini, P., Speciale, A., Cutrona, G., Fais, F., Taiana, E., Fronza, G. (2020, October 28). Heterogeneity of TP53 Mutations and P53 Protein Residual Function in Cancer: Does It Matter? Frontiers in Oncology. Frontiers Media S.A. https://doi.org/10.3389/fonc.2020.593383.

[5] Baugh, E., Ke, H., Levine, A. et al. Why are there hotspot mutations in the TP53 gene in human cancers?. Cell Death Differ 25, 154–160 (2018). https://doi.org/10.1038/cdd.2017.180.

[6] Mantovani, F., Collavin, L. & Del Sal, G. Mutant p53 as a guardian of the cancer cell. Cell Death Differ26, 199–212 (2019). https://doi.org/10.1038/s41418-018-0246-9.

[7] Zhu, G., Pan, C., Bei, J. X., Li, B., Liang, C., Xu, Y., & Fu, X. (2020, November 6). Mutant p53 in Cancer Progression and Targeted Therapies. Frontiers in Oncology. Frontiers Media SA https://doi.org/10.3389/fonc.2020.595187.

[8] Alvarado-Ortiz, E., de la Cruz-López, K. G., Becerril-Rico, J., Sarabia-Sánchez, M. A., Ortiz-Sánchez, E., &García-Carrancá, A. (2021, February 11). Mutant p53 Gain-of-Function: Role in Cancer Development, Progression, and Therapeutic Approaches. Frontiers in Cell and Developmental Biology. Frontiers Media SA https://doi.org/10.3389/fcell.2020.607670.

[9] Zhou, X., Hao, Q., & Lu, H. (2019, April 1). Mutant p53 in cancer therapy-the barrier or the path. Journal of Molecular Cell Biology. Oxford University Press. https://doi.org/10.1093/jmcb/mjy072.

[10] Pavlakis, E., &Stiewe, T. (2020, February 1). p53's extended reach: The mutant p53 secretome. Biomolecules. MDPI AG. https://doi.org/10.3390/biom10020307.

[11] Demir, S., Boldrin, E., Sun, Q., Hampp, S., Tausch, E., Eckert, C., Meyer, L. H. (2020). Therapeutic targeting of mutant p53 in pediatric acute lymphoblastic leukemia. Haematologica, 105(1), 170–181. https://doi.org/10.3324/haematol.2018.199364.

[12] Hassan, M. S., Shaalan, A. A., Dessouky, M. I., Abdelnaiem, A. E., &ElHefnawi, M. (2019). Evaluation of computational techniques for predicting non-synonymous single nucleotide variants pathogenicity. Genomics, 111(4), 869–882. https://doi.org/10.1016/j.ygeno.2018.05.013.

[13] Arshad, S., Ishaque, I., Mumtaz, S., Rashid, M. U., &Malkani, N. (2021). In-Silico Analyses of Non-synonymous Variants in the BRCA1 Gene. Biochemical Genetics. https://doi.org/10.1007/s10528-021-10074-7.

[14] Li, Y., Gordon, M. W., Xu-Monette, Z. Y., Visco, C., Tzankov, A., Zou, D., Young, K. H. (2013). Single nucleotide variation in the TP53 3′ untranslated region in diffuse large B-cell lymphoma treated with rituximab-CHOP: A report from the International DLBCL Rituximab-CHOP Consortium Program. Blood, 121(22), 4529–4540. https://doi.org/10.1182/blood-2012-12-471722.

[15] Poirion, O., Zhu, X., Ching, T. et al. Using single nucleotide variations in single-cell RNA-seq to identify subpopulations and genotype-phenotype linkage. Nat Commun9, 4892 (2018). https://doi.org/10.1038/s41467-018-07170-5.

[16] Almarzooqi, F., Souid, A. K., Vijayan, R., & Al-Hammadi, S. (2021). Novel genetic variants of inborn errors of immunity. PloS one, 16(1), e0245888. https://doi.org/10.1371/journal.pone.0245888.

[17] Alsamri, M. T., Alabdouli, A., Alkalbani, A. M., Iram, D., Tawil, M. I., Antony, P., Vijayan, R., &Souid, A. K. (2021). Genetic variants of small airways and interstitial pulmonary disease in children. Scientific reports, 11(1), 2715. https://doi.org/10.1038/s41598-021-81280-x.

[18] Gyulkhandanyan, A., Rezaie, A. R., Roumenina, L., Lagarde, N., Fremeaux-Bacchi, V., Miteva, M. A., &Villoutreix, B. O. (2020). Analysis of protein missense alterations by combining sequence- and structure-based methods. Molecular Genetics and Genomic Medicine, 8(4). https://doi.org/10.1002/mgg3.1166.

[19] Patil, S., Moafa, I. H., MosaAlfaifi, M., Abdu, A. M., Jafer, M. A., Raju K, L., Sait, S. M. (2020). Reviewing the Role of Artificial Intelligence in Cancer. Asian Pacific Journal of Cancer Biology, 5(4), 189–199. https://doi.org/10.31557/apjcb.2020.5.4.189-199.

[20] Belciug, S. (2020). Pathologist at work. In Artificial Intelligence in Cancer (pp. 161–186). Elsevier. https://doi.org/10.1016/b978-0-12-820201-2.00003-9.

[21] Ioannidis, N. M., Rothstein, J. H., Pejaver, et. al (2016). REVEL: An Ensemble Method for Predicting the Pathogenicity of Rare Missense Variants. American journal of human genetics, 99(4), 877–885. https://doi.org/10.1016/j.ajhg.2016.08.016.

[22] Niroula, A., &Vihinen, M. (2019). How good are pathogenicity predictors in detecting benign variants? PLoS Computational Biology, 15(2). https://doi.org/10.1371/journal.pcbi.1006481

[23] LYRUS: A Machine Learning Model for Predicting the Pathogenicity of Missense Variants.

[24] Jiaying Lai, Jordan Yang, Ece D. GamsizUzun, Brenda M. Rubenstein, Indra Neil Sarkar.

[25] Gornale, S. S., Kumar, S., Siddalingappa, R., & Mane, A. (2022). Gender Classification Based on Online Signature Features using Machine Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 10(2), 260–268.

[26] Tan, H., Bao, J., & Zhou, X. (2012). A novel missense-mutation-related feature extraction scheme for 'driver' mutation identification. Bioinformatics (Oxford, England), 28(22), 2948–2955. ttps://doi.org/10.1093/bioinformatics/bts558.

[27] Richards, S., Aziz, N., Bale, S., Bick, D., Das, S., Gastier-Foster, J., Rehm, H. L. (2015). Standards and guidelines for the interpretation of sequence variants: A joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. Genetics in Medicine, 17(5), 405–424. https://doi.org/10.1038/gim.2015.30.

[28] SIFT ref: Ng, Pauline C, and Steven Henikoff. "SIFT: Predicting amino acid changes that affect protein function." Nucleic acids research vol. 31,13 (2003): 3812-4. doi:10.1093/nar/gkg509.

[29] Provean ref: Choi Y, Sims GE, Murphy S, Miller JR, Chan AP. Predicting the functional effect of amino acid substitutions and indels. PLoS One. 2012;7(10):e46688. doi: 10.1371/journal.pone.0046688. Epub 2012 Oct 8. PMID: 23056405; PMCID: PMC3466303.

[30] Mutassessor ref: Boris Reva, YevgeniyAntipin, Chris Sander, Predicting the functional impact of protein mutations: application to cancer genomics, Nucleic Acids Research, Volume 39, Issue 17, 1 September 2011, Page e118, https://doi.org/10.1093/nar/gkr407.

[31] Siddalingappa, R., & Kanagaraj, S. (2022). K-nearest-neighbor algorithm to predict the survival time and classification of various stages of oral cancer: a machine learning approach. F1000Research, 11, 70. https://doi.org/10.12688/f1000research.75469.1.

[32] Gewers, F. L., Ferreira, G. R., De Arruda, H. F., Silva, F. N., Comin, C. H., Amancio, D. R., & Costa, L. D. F. (2021). Principal component analysis: A natural approach to data exploration. ACM Computing Surveys, 54(4). https://doi.org/10.1145/3447755.

[33] Fernandes, A. A. T., Filho, D. B. F., da Rocha, E. C., & da Silva Nascimento, W. (2020). Read this paper if you want to learn logistic regression. Revista de Sociologia e Politica, 28(74), 1/1-19/19. https://doi.org/10.1590/1678-987320287406EN.

[34] Gornale, S., Kumar, S., Siddalingappa, R., & Hiremath, P. S. (2022). Survey on Handwritten Signature Biometric Data Analysis for Assessment of Neurological Disorder using Machine Learning Techniques. Transactions on Machine Learning and Artificial Intelligence, 10(2), 27–60. https://doi.org/10.14738/tmlai.102.12210.

[35] Rashmi, S., Hanumanthappa, M., &Jyothi, N. M. (2016). Text-to-Speech translation using Support Vector Machine, an approach to find a potential path for human-computer speech synthesizer. In Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016 (pp. 1311–1315). Presses PolytechniquesEtUniversitairesRomandes. https://doi.org/10.1109/WiSPNET.2016.7566349.

[36] Shaheen, M., Zafar, T., & Ali Khan, S. (2020). Decision tree classification: Ranking journals using IGIDI. Journal of Information Science, 46(3), 325–339. https://doi.org/10.1177/0165551519837176.

[37] Schonlau, M., &Zou, R. Y. (2020). The random forest algorithm for statistical learning. Stata Journal, 20(1), 3–29. https://doi.org/10.1177/1536867X20909688.

[38] Rashmi Siddalingappa and Sekar Kanagaraj, "Anomaly Detection on Medical Images using Autoencoder and Convolutional Neural Network" International Journal of Advanced Computer Science and Applications(IJACSA), 12(7), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120717.

[39] Hamroun, D., Kato, S., Ishioka, C., Claustres, M., Béroud, C., &Soussi, T. (2006, January). The UMD TP53 database and website: Update and revisions. Human Mutation. https://doi.org/10.1002/humu.20269.