ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **X**, Number X, XXXX, XXX–XXX

Supplementary material of the paper "Deep Learning-Based Prediction Models for the Detection of Vitamin D Deficiency and 25-Hydroxyvitamin D Levels Using Complete Blood Count Tests"

Uğur Engin EŞSİZ[1], Çiğdem İnan ACI[2,*], Esra SARAÇ[3], and Mehmet ACI[2]

[1]Department of Industrial Engineering, Çukurova University, Adana, Turkey
[2]Department of Computer Engineering, Mersin University, Mersin, Turkey
[3]Department of Computer Engineering, Adana Alparslan Türkeş Science and Technology University, Adana, Turkey

E-mails: enginessiz@gmail.com, caci@mersin.edu.tr*,esarac@atu.edu.tr, maci@mersin.edu.tr
* Corresponding author

This PDF file contains the tables and the figures created for the Deep Learning-Based Prediction Models for the Detection of Vitamin D Deficiency and 25-Hydroxyvitamin D Levels Using Complete Blood Count Tests titled article published in the Romanian Journal of Information Science and Technology. You can access the article via this link: (the link will be added after publication)

Distribution of CBC tests in terms of VitD status.

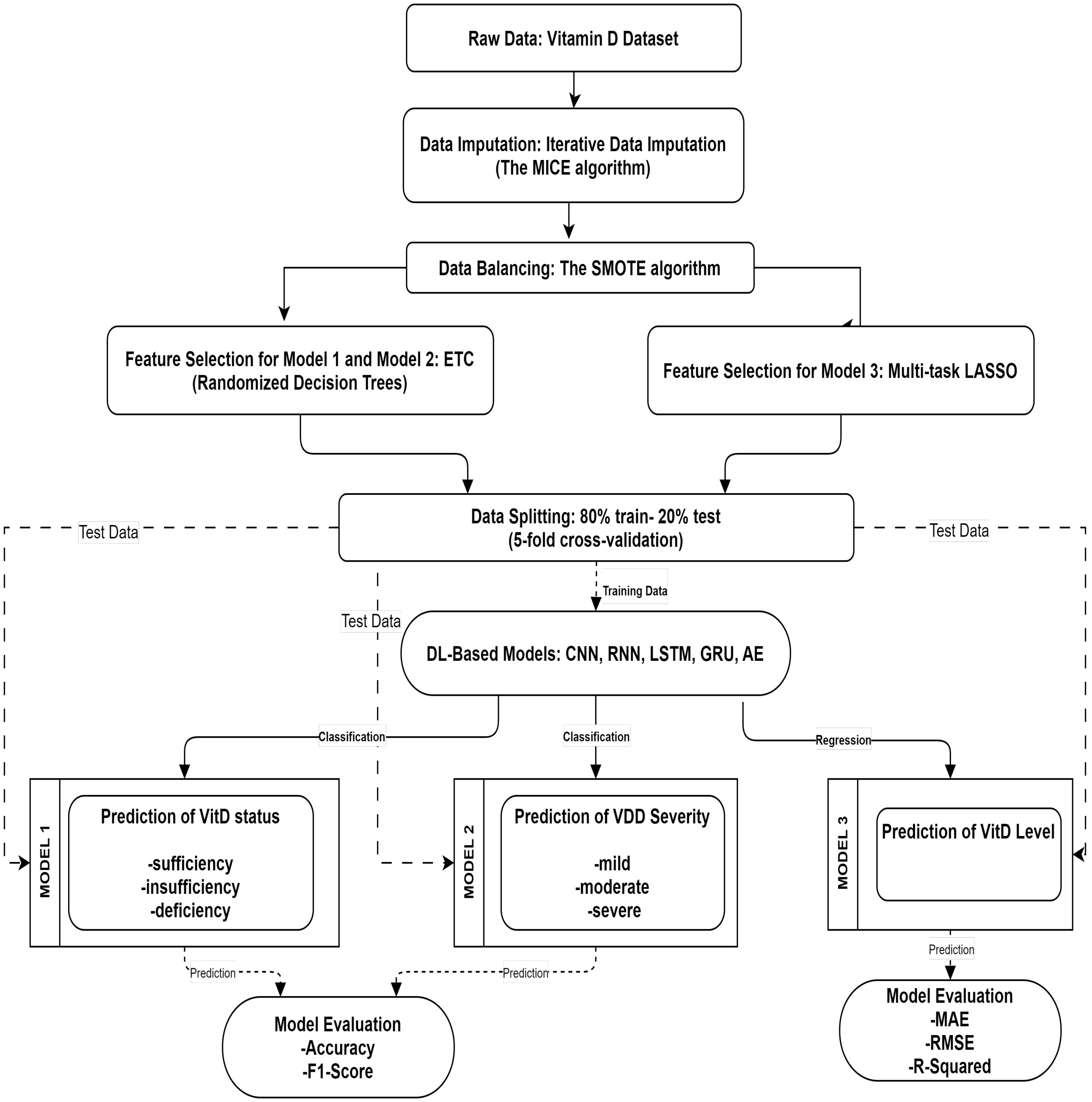
25(OH)D Level	Diagnosis	Frequency			Percentage (%)
		# of patients	Male	Female	
>=30 ng/mL	VitD Sufficiency	255	59	196	28.11
21-29 ng/mL	VitD Insufficiency	185	104	81	20.40
<=20 ng/mL	VitD Deficiency	467	235	232	51.49
	Total	907	398	509	100

Distribution CBC tests in terms of patient's VDD.

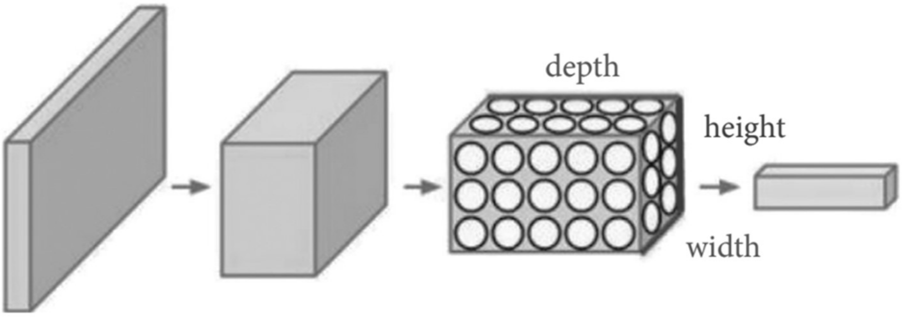
25(OH)D Level	Diagnosis	Frequency			Percentage (%)
		# of patients	Male	Female	
11-20 ng/mL	Mild VDD	260	159	101	55.68
5-10 ng/mL	Moderate VDD	182	72	110	38.97
<5 ng/mL	Severe VDD	25	4	21	5.35
	Total	467	235	232	100

Age and gender distribution of patients.

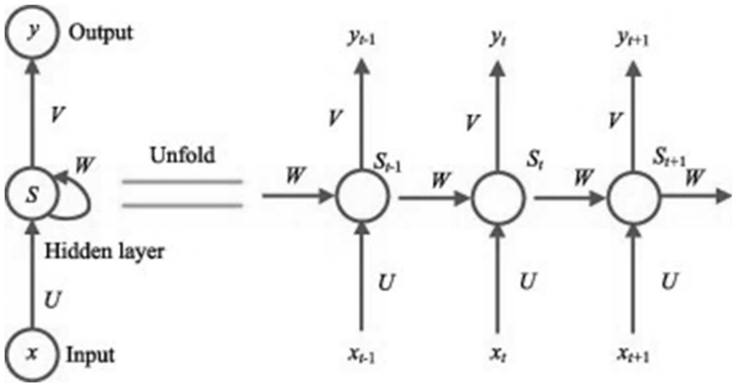
Age	Frequency	Percentage (%)
Child (<=10)	40	4.40
Adolescent (11-17)	76	8.50
Adult (18-64)	781	86.00
Elderly (65+)	10	1.10
Total	907	100


Raw Data: Vitamin D Dataset
Data Imputation: Iterative Data Imputation (The MICE algorithm)
Data Balancing: The SMOTE algorithm
Feature Selection for Model 1 and Model 2: ETC (Randomized Decision Trees)
Feature Selection for Model 3: Multi-task LASSO
Data Splitting: 80% train- 20% test (5-fold cross-validation)
Test Data
Test Data
Test Data
Training Data
DL-Based Models: CNN, RNN, LSTM, GRU, AE
Classification
Classification
Regression
MODEL 1
Prediction of VitD status
-sufficiency
-insufficiency
-deficiency
MODEL 2
Prediction of VDD Severity
-mild
-moderate
-severe
MODEL 3
Prediction of VitD Level
Prediction
Prediction
Prediction
Model Evaluation
-Accuracy
-F1-Score
Model Evaluation
-MAE
-RMSE
-R-Squared

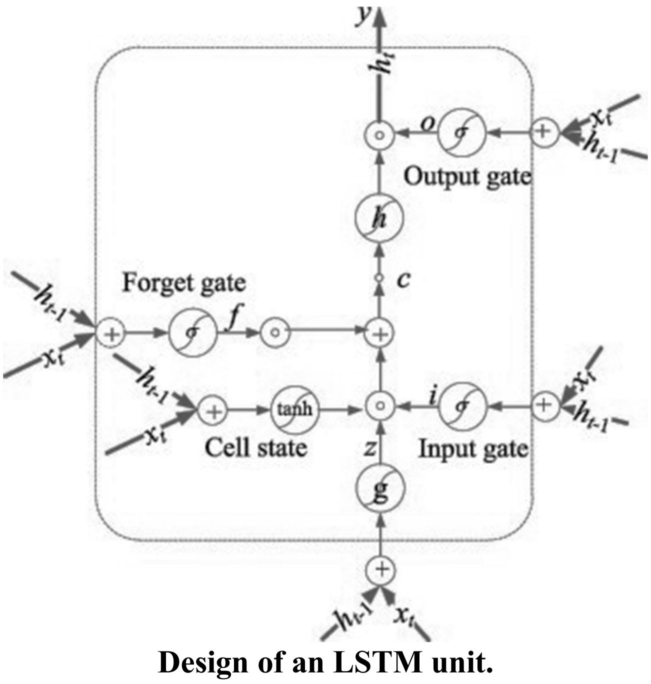
Block diagram of the DL-based prediction models.



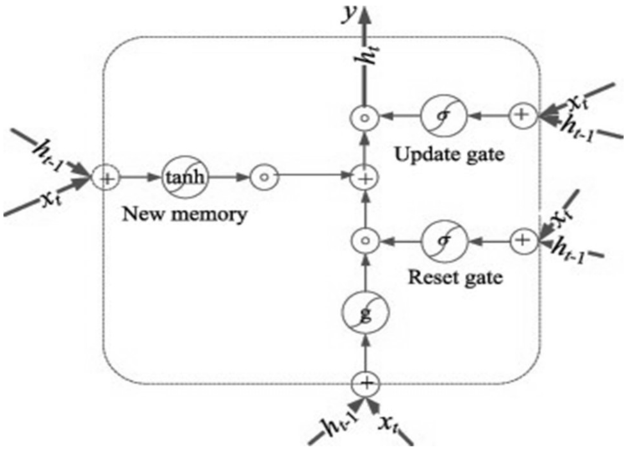
Schematic diagram of a CNN.



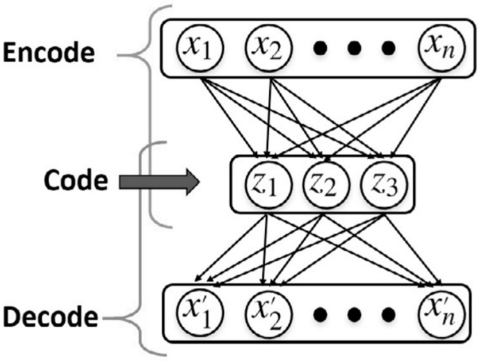
.The basic structure of an RNN model.



Design of an LSTM unit.



Structure of a GRU.



Structure of an AE.

Pseudo-code for Model 1 and Model 2

```
Input: An instance of a DL-based classifier method, DL_Clf
Input: Training dataset, $X_{train}$
Input: Test dataset, $X_{test}$
Input: Necessary parameters for data splitting, param
Input: A value for cross validation, kfold
Input: A parameter-set for the MICE algorithm, imputValue
Input: A parameter-set for data balancing technique, balanceValue
Input: A parameter-set for feature selection technique, featureValue
Input: A parameter-set for grid search technique, gridValue
Input: Number of times model execution, E
Output: score: The assessment metrics are computed on the test dataset

Begin
Pre-processing
($X_{train}$,$X_{test}$)←MICE($X_{train}$,$X_{test}$,imputValue)
($X_{train}$,$X_{test}$)←SMOTE($X_{train}$,$X_{test}$,balanceValue)
($X_{train}$,$X_{test}$)←ETC($X_{train}$,$X_{test}$,featureValue)

Train
($X_{train\_input}$,$X_{train\_output}$)←split($X_{train\_input}$,$X_{train\_output}$, param)
model←calculate(DL_Clf,gridValue,kfold, $X_{train\_input}$,$X_{train\_output}$)
error←calculate_error(model)
while model has not converged do
  while epoch←1 to E do
    Update model parameters, $w_i \leftarrow w_i - \eta \frac{\partial error}{\partial w}$
  end
end

Test
Evaluate trained model with test dataset
score←computeMetrics(model,$X_{model\_output}$,$X_{test\_output}$,'accuracy','f1_score')
return score
End
```

Pseudo-code for Model 3

```
Input: An instance of a DL-based regression method, DL_Reg
Input: Training dataset, $X_{train}$
Input: Test dataset, $X_{test}$
Input: Necessary parameters for data splitting, param
Input: A value for cross validation, kfold
Input: A parameter-set for the MICE algorithm, imputValue
Input: A parameter-set for data balancing technique, balanceValue
Input: A parameter-set for feature selection technique, featureValue
Input: A parameter-set for grid search technique, gridValue
Input: Number of times model execution, E
Output: score: The assessment metrics are computed on the test dataset

Begin
Pre-processing
($X_{train}$,$X_{test}$)←MICE($X_{train}$,$X_{test}$,imputValue)
($X_{train}$,$X_{test}$)←SMOTE($X_{train}$,$X_{test}$,balanceValue)
($X_{train}$,$X_{test}$)←Multi_task_LASSO($X_{train}$,$X_{test}$,featureValue)

Train
($X_{train\_input}$,$X_{train\_output}$)←split($X_{train\_input}$,$X_{train\_output}$,param)
model←calculate(DL_Reg,gridValue,kfold,$X_{train\_input}$,$X_{train\_output}$)
error←calculate_error(model)
while model has not converged do
  while epoch←1 to E do
    Update model parameters, $w_i \leftarrow w_i - \eta \frac{\partial error}{\partial w}$
  end
end

Test
Evaluate trained model with test dataset
score←computeMetrics(model,$X_{model\_output}$,$X_{test\_output}$,'mae','rmse','r-squared')
return score
End
```

Pseudo-code for Multivariate Imputation by Chained Equations (MICE) algorithm (Iterative Data Imputation)

```
Define $Y$ as a $nxp$ data matrix where rows represent samples and columns represent variables.
Data: Incomplete dataset $Y = (Y^{obs}, Y^{mis})$
Result: Incomplete dataset $Y^{T} = (Y^{obs}, Y^{mis,T})$ at iteration $T$
Define $Y_j$ as the $j^{th}$ feature column of $Y$ where $Y_j = (Y_j^{obs}, Y_j^{mis})$
for $j$ ←1 to $p$ do
  imputation model for incomplete variable $Y_j \leftarrow P(Y_j | Y_{-j}, \theta_j)$
  starting imputations $Y_j^{mis,0}$ ←draws from $Y_j^{obs}$
Define $Y_{-j}^{t} = (Y_1^t, Y_2^t, \dots, Y_{j-1}^t, Y_{j+1}^{t-1}, \dots, Y_{p-1}^{t-1}, Y_p^{t-1})$ where $Y_j^t$ is the $j^{th}$ feature at iteration $t$
for t←1 to $T$ do
  for $j$ ←1 to $p$ do
    $\theta_j^t$ draw from posterior $P(\theta_j | Y_j^{obs}, Y_{-j}^{t})$
    $Y_j^{mis,t}$ draws from posterior predictive $P(Y_j^{mis} | Y_{-j}^{t}, \theta_j^t)$
return $\mathrm{Y}^{\mathrm{T}}$
```

Pseudo-code for the Synthetic Minority Oversampling Technique (SMOTE)

```
Input: Minority data $D^{(t)} = \{x_i \in X\}$ where $i = 1,2,..,T$
       Number of minority instances $(T)$, SMOTE percentage$(N)$,
       number of nearest neighbors $(k)$
for $i = 1,2,..,T$ do
       Find the $k$ nearest (minority class) neighbors of $x_i$
       $\widehat{\mathrm{N}} = \lfloor \mathrm{N}/100 \rfloor$
       while $\widehat{\mathrm{N}} \neq 0$ do
              Select one of the $k$ nearest neighbors, call this $\bar{x}$
              Select a random number $\alpha \in [0,1]$
              $\hat{\mathrm{x}} = x_i + \alpha(\bar{x} - x_i)$
              Append $\hat{\mathrm{x}}$ to $S$
              $\widehat{\mathrm{N}} = \widehat{\mathrm{N}} - 1$
       end while
end for
Output: Return synthetic data $S$
```

Pseudo-code for Randomized Decision Trees (Extra Trees Classifier) (ETC)

```
Given $(x1, y1), (x2, y2), ..., (xn, yn)$ with feature space $F$ where $xi \in X$ and $yi \in \Omega$
Given the number of decision trees M and max depth of each tree $max\_depth$
procedure Train $((x1, y1), (x2, y2), ..., (xn, yn))$
   $ETC \leftarrow \{\}$
   for $i$ from 1 to $M$ do
       $Ti \in X \rightarrow \Omega$
       while ( do $depth(T) < max\_depth$)
       Randomly select $Xi \subset X$ without replacement
       Randomly select feature $f \in F$
         Use $f$ as the node to construct tree
       end while
       $ETC = ETC \cup \{Ti\}$
    end for
    return $ETC$
end procedure
procedure Test($x$)
    for $i$ from 1 to $M$ do
    Select decision tree $Ti$ from $ETC$
    $yi \leftarrow Ti(x)$
    end for
    $y = \frac{\sum_{i=1}^{M} y_i}{M}$
return $y$
end procedure
```

LASSO Regularization (L1)

$$loss = \sum_{i=0}^{n} (y_i - y')^2$$

$$loss = \sum_{i=0}^{n} (y_i - X_i\beta)^2$$

$$L1 = \frac{1}{n}\sum_{i=0}^{n} (y_i - X_i\beta)^2 + \frac{\lambda}{n}\sum_{j=0}^{m} |\beta|$$

where y_i actual, y' predicted, X input, β co-efficient, λ regularization parameter.

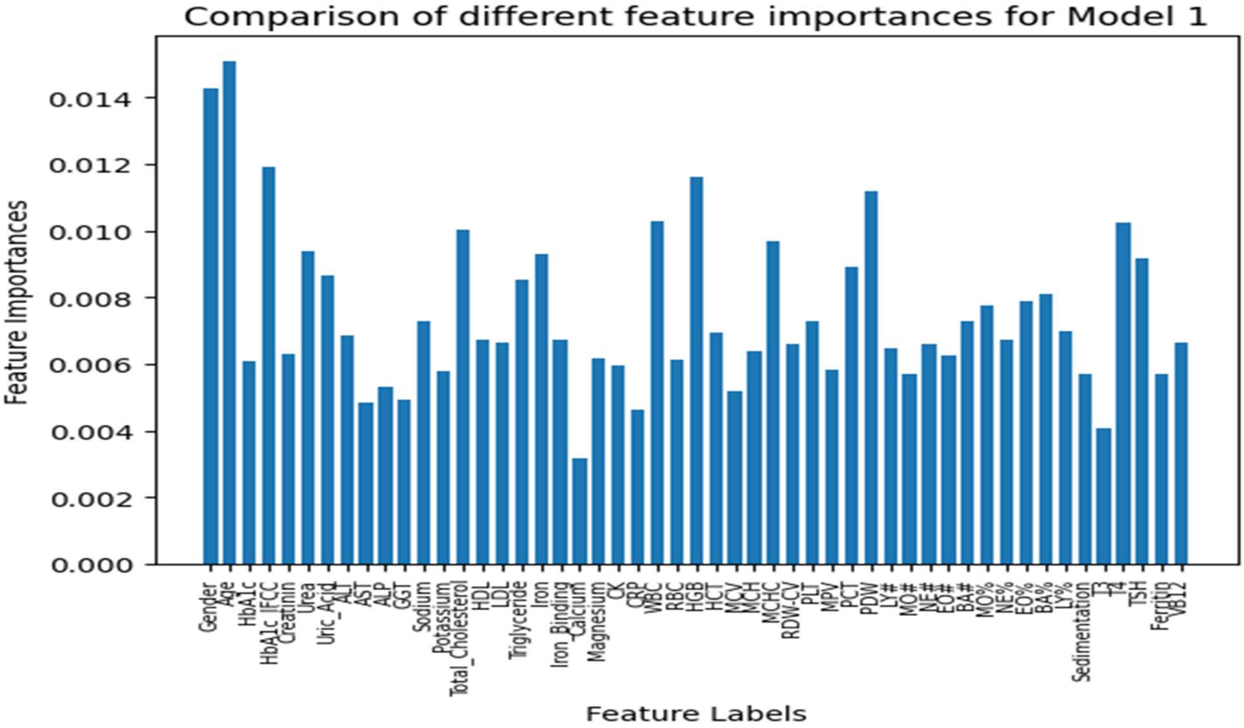
Parameter settings for the methods.

Method	Parameter: Value	
	Model 1- Model 2	Model 3
CNN	Activation Function: ReLU and Sigmoid, Loss: Binary Crossentr, Epochs: 100, Batch Size: 32	
	Dense: 256 / 128 / 25 / 3 neurons Optimizer: Rmsprop	Dense: 256 / 128 / 25 / 1 neurons Optimizer: Adam
RNN	Activation Function: Sigmoid, Recurrent Initializer=orthogonal, Embedding: 20000, Loss: Binary Crossentr, Epochs: 100, Batch Size: 32	
	Output: 64 / 25 / 3 neurons Optimizer: Rmsprop	Output: 64 / 25 / 1 neurons Optimizer: Adam
LSTM	Activation Function: Tanh, Recurrent Activation=sigmoid, Embedding: 20000, Loss: Binary Crossentr, Epochs: 100, Batch Size: 32	
	Output: 64 / 25/ 3 neurons Optimizer: Rmsprop	Output: 64 / 25 / 1 neurons Optimizer: Adam
GRU	Activation Function: Tanh, Recurrent Activation=sigmoid, Embedding: 20000, Loss: Binary Crossentr, Epochs: 100, Batch Size: 32	
	Output: 64 / 25 / 3 neurons Optimizer: Rmsprop	Output: 64 / 25 / 1 neurons Optimizer: Adam
AE	Activation Function: ReLU , Loss: MSE, Epochs: 100, Batch Size: 32	
	Encoder: 50/25/3 Decoder: 3/25/50 Output: 3 Optimizer: Rmsprop	Encoder: 50/25/3 Decoder: 3/25/50 Output: 1 Optimizer: Adam
SMOTE	Model: DecisionTreeClassifier(), k_neighbors: 2	sampling_strategy: 0.1
ETC	n_estimators: 10, criterion:'entropy', max_features: 3	-
Multi-task LASSO	-	regularization parameter (λ): 0.5
Grid Search	estimator: RandomForestClassifier, max_features: ['auto', 'sqrt', 'log2'], criterion :['gini', 'entropy']	

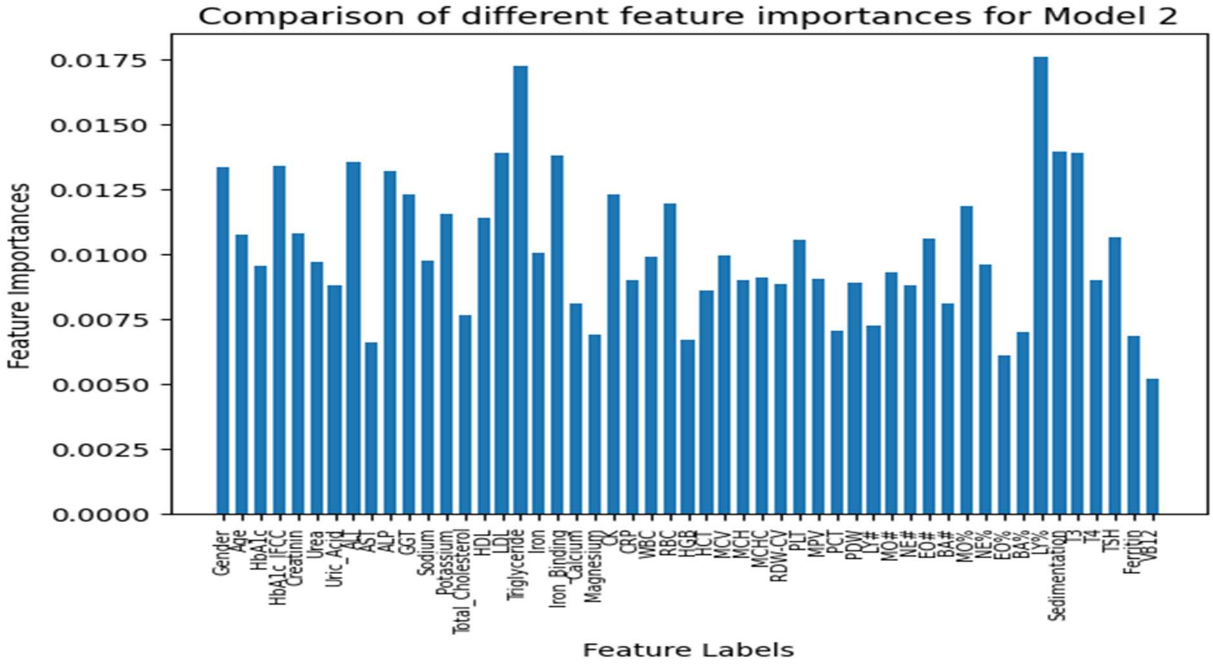
The best attribute indexes for the models.

Model	Feature Selection	Top-25 feature indexes
Model 1	ETC	[0 1 3 5 6 7 11 13 16 17 23 25 26 29 31 33 34 39 40 42 43 44 47 48 50]
Model 2	ETC	[0 1 3 4 5 7 9 10 11 12 14 15 16 18 21 24 27 31 38 40 41 44 45 46 48]
Model 3	Multi-Task LASSO	[0 1 2 3 4 8 11 12 14 15 16 17 18 19 22 23 24 27 28 29 35 45 46 48 49]

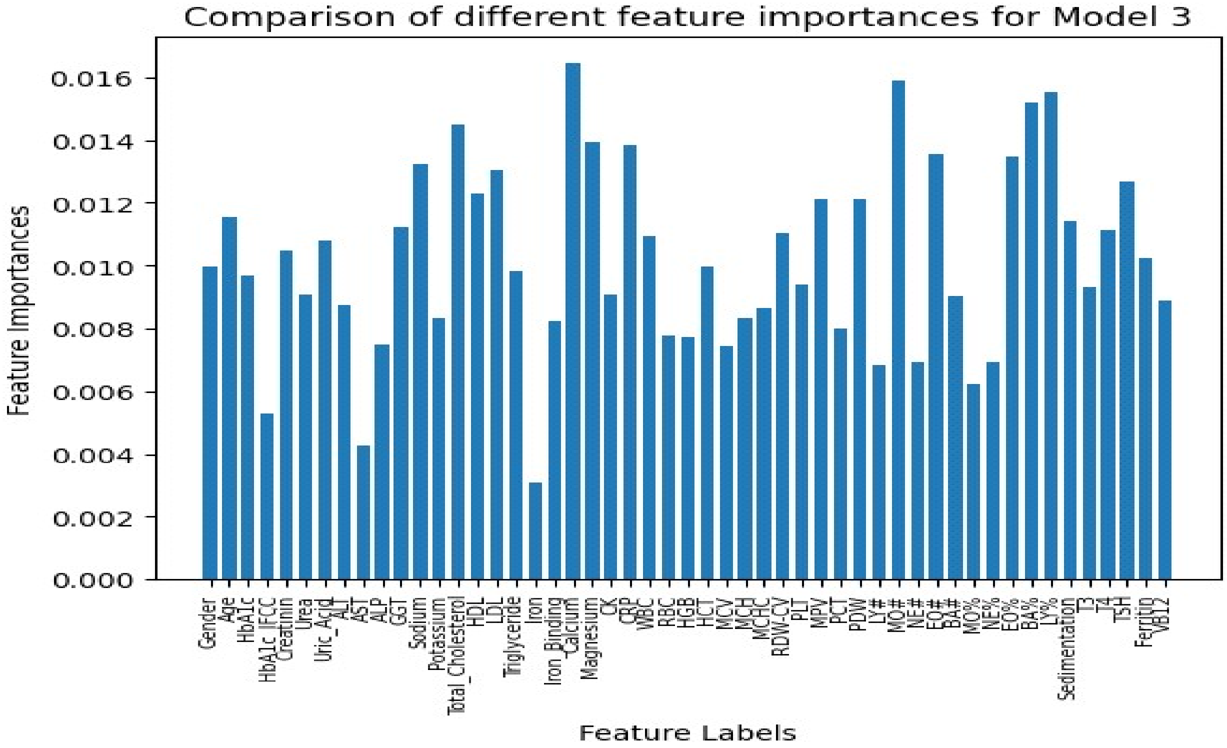
Feature Importance Distribution of the Models



Feature importance distribution of the Model 1.

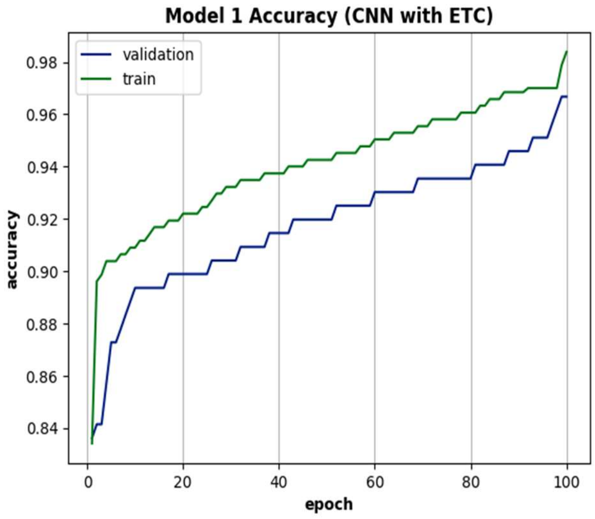


Feature importance distribution of the Model 2.

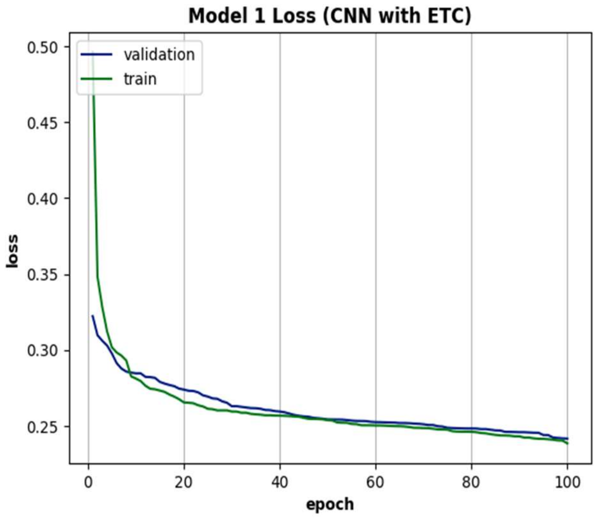


Feature importance distribution of the Model 3.

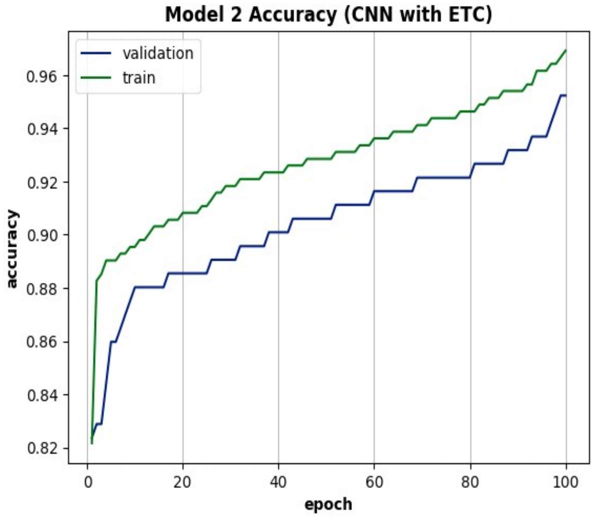
Convergence Plots of the Models



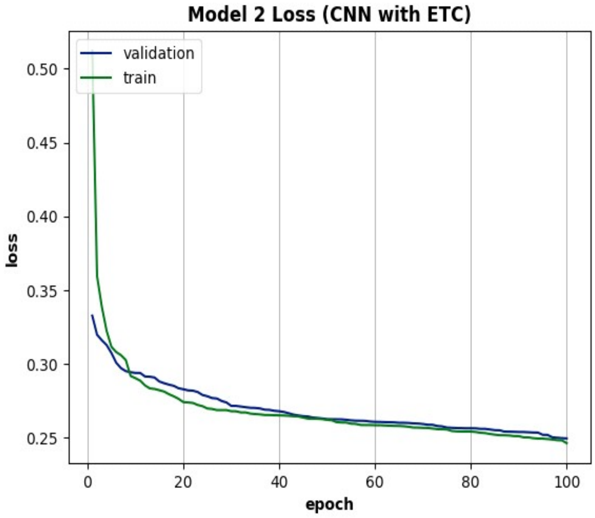
Model 1 Accuracy (CNN with ETC)



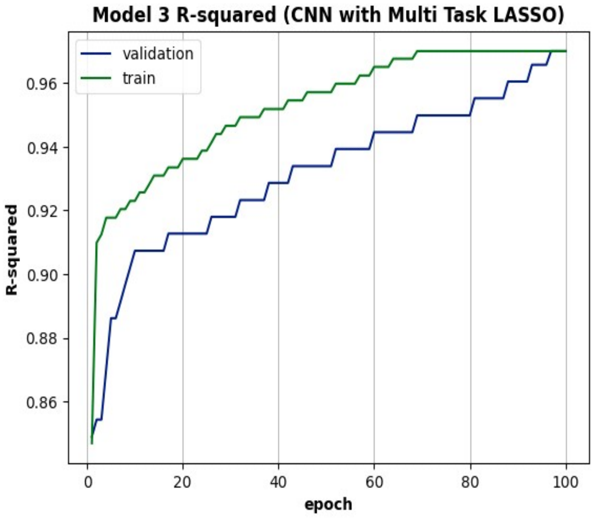
Model 1 Loss (CNN with ETC)



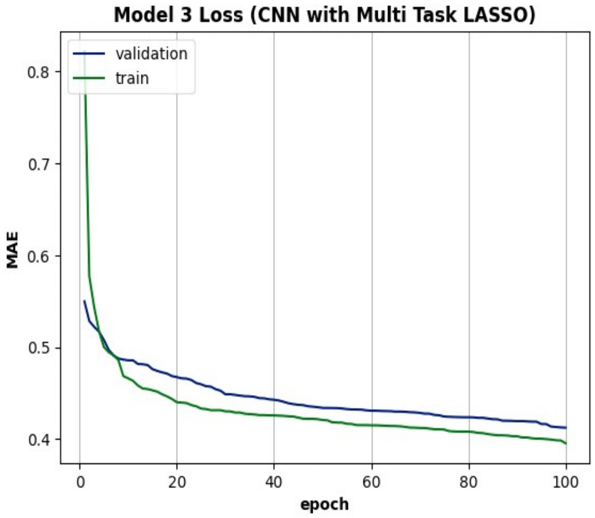
Model 2 Accuracy (CNN with ETC)



Model 2 Loss (CNN with ETC)



Model 3 R-squared (CNN with Multi Task LASSO)



Model 3 Loss (CNN with Multi Task LASSO)