

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 EŞSİZ¹, Çiğdem İnan ACI^{2,*}, Esra SARAÇ³, and Mehmet ACI²

¹Department of Industrial Engineering, Çukurova University, Adana, Turkey

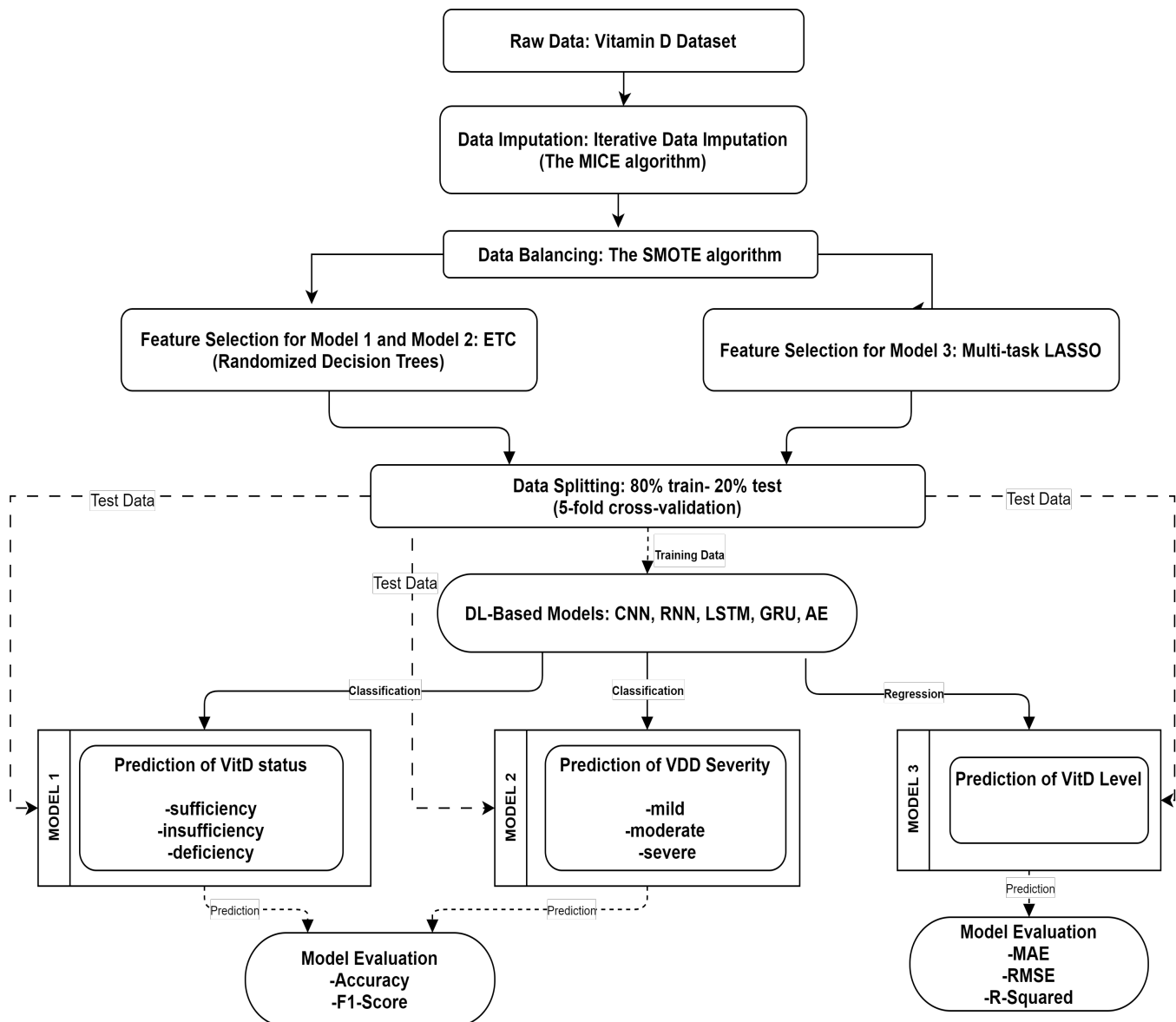
²Department of Computer Engineering, Mersin University, Mersin, Turkey

³Department 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: [\(the link will be added after publication\)](#)



Block diagram of the DL-based prediction models.

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

Define Y as a $n \times p$ 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 $j \leftarrow 1$ to p **do**

 imputation model for incomplete variable $Y_j \leftarrow P(Y_j | Y_{-j}, \theta_j)$

 starting imputations $Y_j^{mis,0} \leftarrow \text{draws from } Y_j^{obs}$

Define $Y_{-j}^t = (Y_1^t, Y_2^t, \dots, Y_{j-1}^t, Y_{j+1}^t, \dots, Y_{p-1}^t, Y_p^t)$ where Y_j^t is the j^{th} feature at iteration t

for $t \leftarrow 1$ to T **do**

for $j \leftarrow 1$ to p **do**

θ_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 Y^T

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

Input: Minority data $D^{(t)} = \{x_i \in X\}$ where $i = 1, 2, \dots, T$

 Number of minority instances (T), SMOTE percentage(N),

 number of nearest neighbors (k)

for $i = 1, 2, \dots, T$ **do**

 Find the k nearest (minority class) neighbors of x_i

$\hat{N} = \lfloor N/100 \rfloor$

while $\hat{N} \neq 0$ **do**

 Select one of the k nearest neighbors, call this \bar{x}

 Select a random number $\alpha \in [0, 1]$

$\hat{x} = x_i + \alpha(\bar{x} - x_i)$

 Append \hat{x} to S

$\hat{N} = \hat{N} - 1$

end while

end for

Output: Return synthetic data S

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

Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ with feature space F where $x_i \in X$ and $y_i \in \Omega$

Given the number of decision trees M and max depth of each tree max_depth

procedure Train $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$

$ETC \leftarrow \{\}$

for i from 1 to M **do**

$T_i \in X \rightarrow \Omega$

while (**do** $depth(T) < max_depth$)

 Randomly select $X_i \subset X$ without replacement

 Randomly select feature $f \in F$

 Use f as the node to construct tree

end while

$ETC = ETC \cup \{T_i\}$

end for

return ETC

end procedure

procedure Test (x)

for i from 1 to M **do**

 Select decision tree T_i from ETC

$y_i \leftarrow T_i(x)$

end for

$y = \frac{\sum_{i=1}^M y_i}{M}$

return y

end procedure

LASSO Regularization (L1)

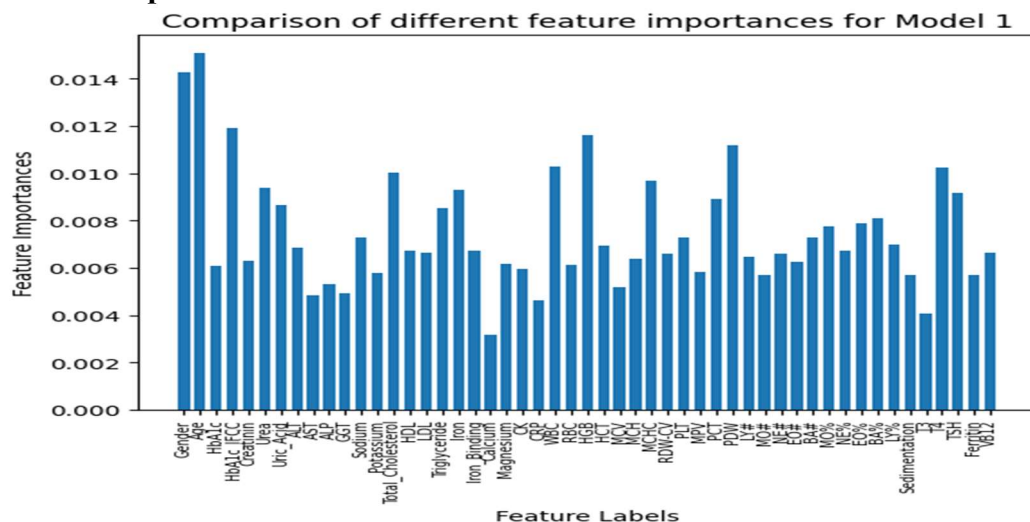
$$loss = \sum_{i=0}^n (y_i - y')^2$$

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

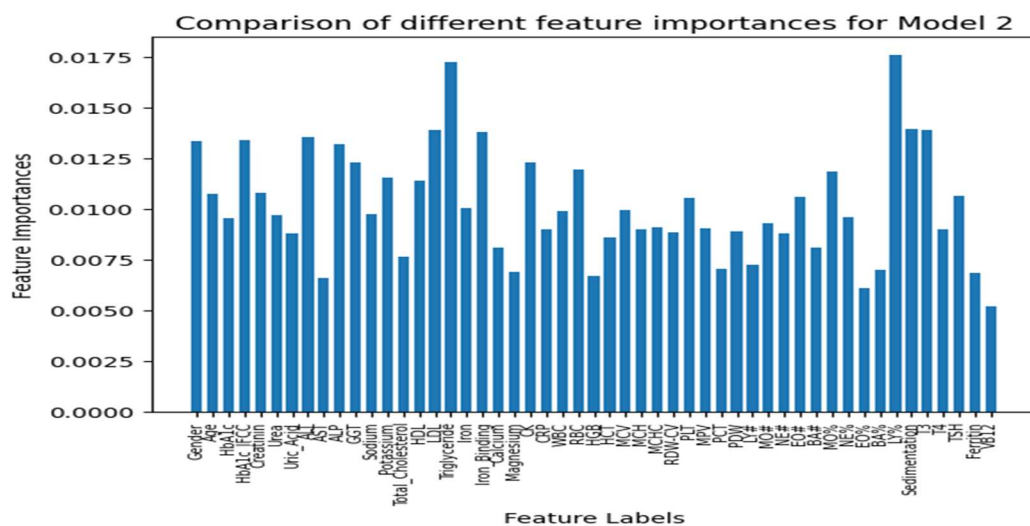
$$L1 = \frac{1}{n} \sum_{i=0}^n (y_i - X_i \beta)^2 + \frac{\lambda}{n} \sum_{j=0}^m |\beta|$$

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

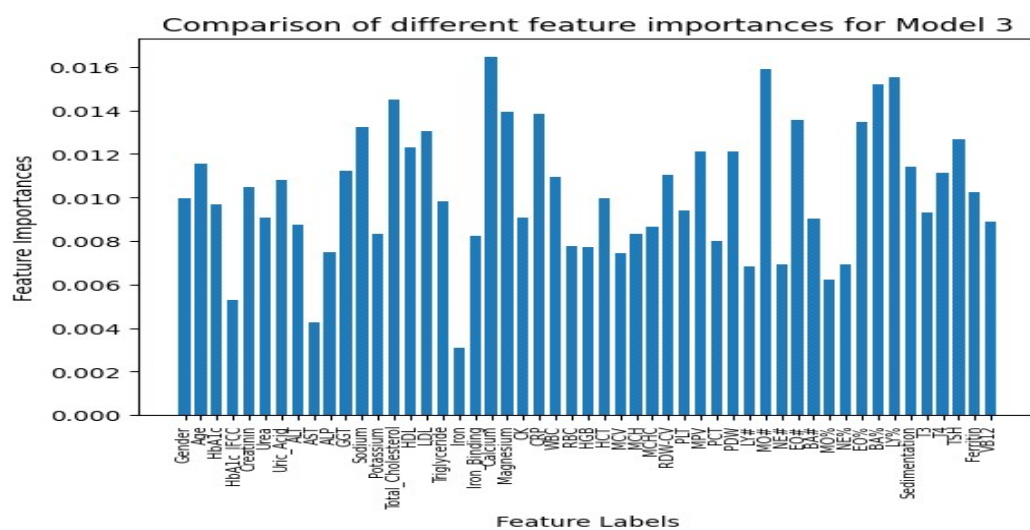
Feature Importance Distribution of the Models



Feature importance distribution of the Model 1.



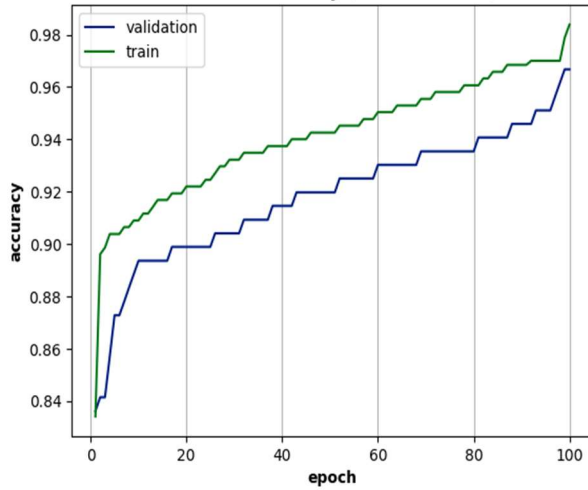
Feature importance distribution of the Model 2.



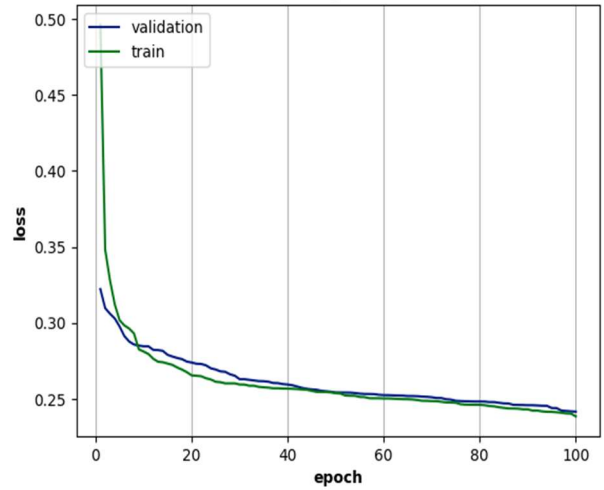
Feature importance distribution of the Model 3.

Convergence Plots of the Models

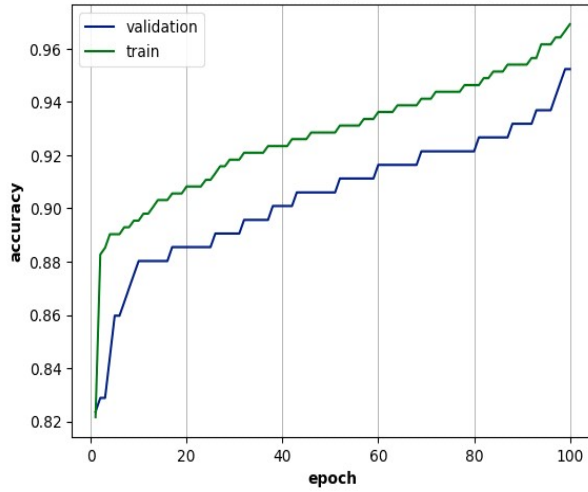
Model 1 Accuracy (CNN with ETC)



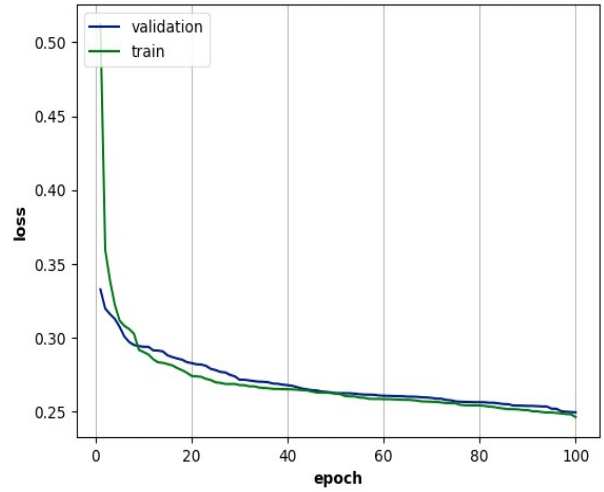
Model 1 Loss (CNN with ETC)



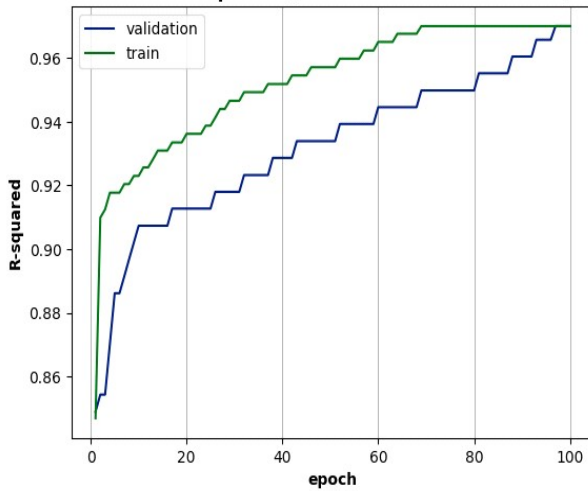
Model 2 Accuracy (CNN with ETC)



Model 2 Loss (CNN with ETC)



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



Model 3 Loss (CNN with Multi Task LASSO)

