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

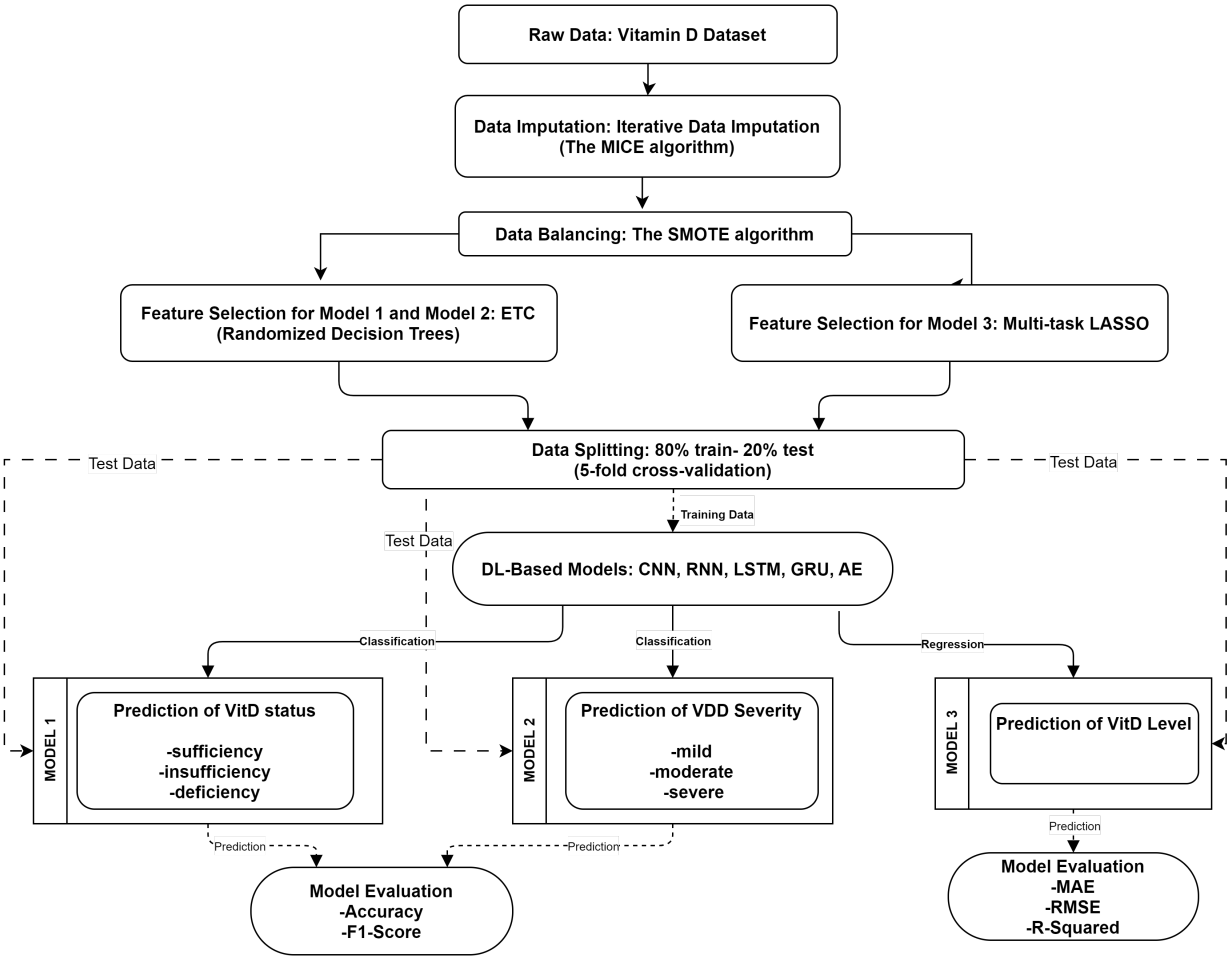
Supplementary material of the paper "Deep Learning-Based Prediction Models for the Detection of Vitamin D Deficiency and 25-Hydroxyvitamin D Levels Using Complete Blood Count Tests"

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)



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 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 ← P(Y_j | Y_{-j}, θ_j)
  starting imputations Y_j^{mis,0} ←draws from Y_j^obs
Define Y_{-j}^t = (Y_1^t, Y_2^t, …, Y_{j-1}^t, Y_{j+1}^{t-1}, …, 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
      θ_j^t draw from posterior P(θ_j | Y_j^obs, Y_{-j}^t)
      Y_j^{mis,t} draws from posterior predictive P(Y_j^mis | Y_{-j}^t, θ_j^t)
return Y^T
```

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

```
Input: Minority data D^(t) = {x_i ∈ 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
       N̂ = ⌊N/100⌋
       while N̂ ≠ 0 do
              Select one of the k nearest neighbors, call this x̄
              Select a random number α ∈ [0,1]
              x̂ = x_i + α(x̄ − x_i)
              Append x̂ to S
              N̂ = N̂ − 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 ∈ X and
yi ∈ Ω
Given the number of decision trees M and max depth of each tree
max_depth
procedure Train((x1,y1),(x2,y2),...,(xn,yn))
   ETC ← {}
   for i from 1 to M do
       Ti ∈ X → Ω
       while ( do depth(T) < max_depth)
       Randomly select Xi ⊂ X without replacement
       Randomly select feature f ∈ F
         Use f as the node to construct tree
       end while
       ETC = ETC ∪ {Ti}
    end for
    return ETC
end procedure
procedure Test(x)
    for i from 1 to M do
    Select decision tree Ti from ETC
    yi ← Ti(x)
    end for
    y = (∑_{i=1}^{M} y_i) / M
return y
end procedure
```

LASSO Regularization (L1)

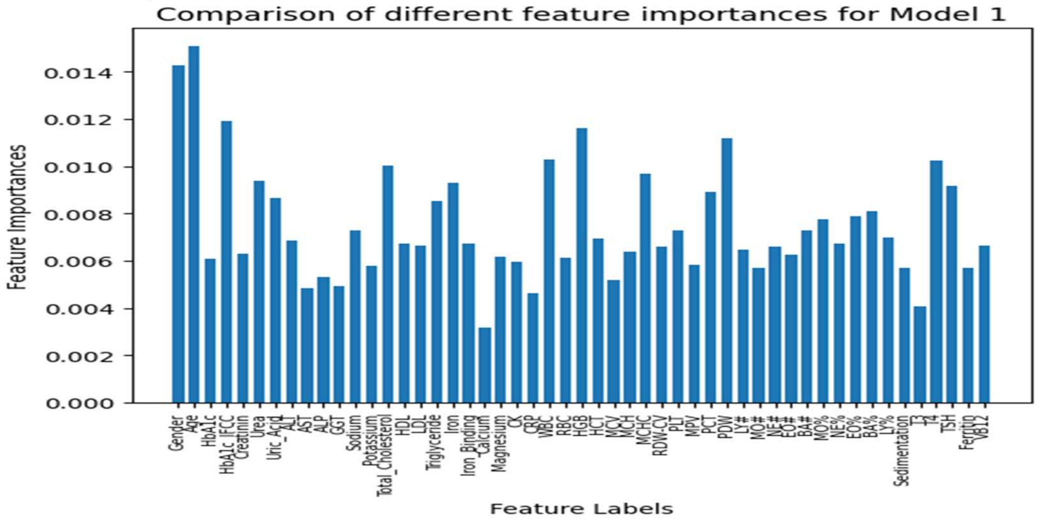
$$loss = \sum_{i=0}^{n} (y_i - y')^2$$

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

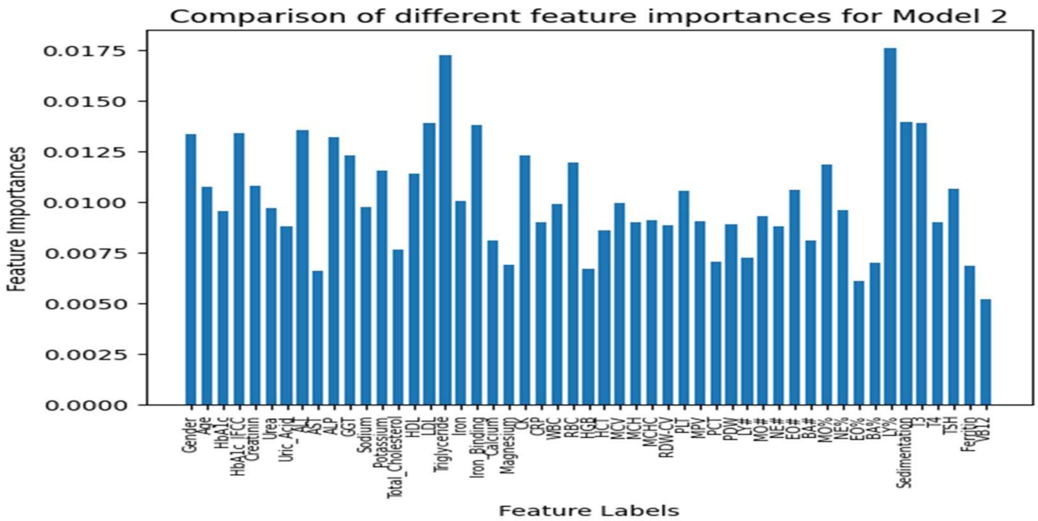
$$L1 = \frac{1}{n}\sum_{i=0}^{n} (y_i - X_i\beta)^2 + \frac{\lambda}{n}\sum_{j=0}^{m} |\beta|$$

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

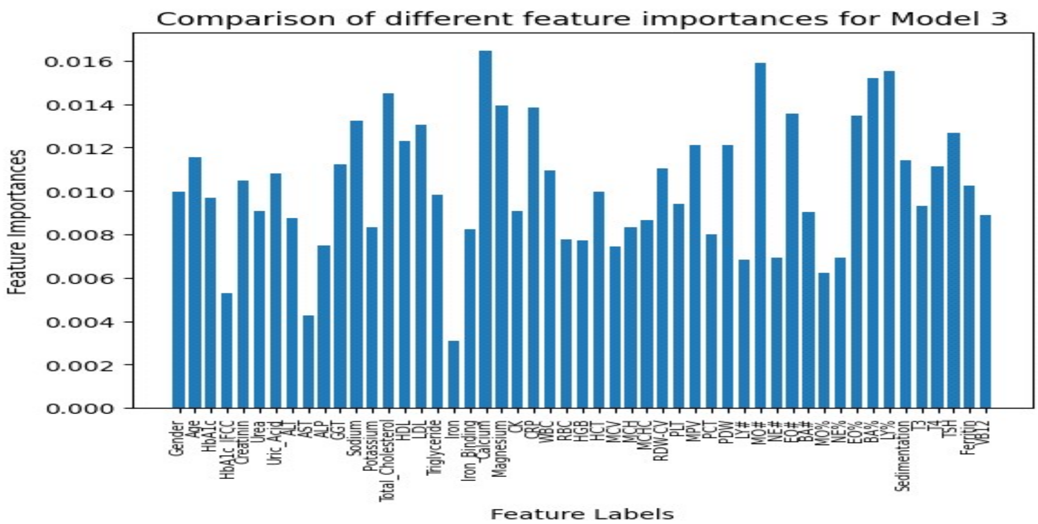
Feature Importance Distribution of the Models



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