**International Journal of Data Science and Analysis**
2022; 8(2): 47-58
http://www.sciencepublishinggroup.com/j/ijdsa
doi: 10.11648/j.ijdsa.20220802.15
ISSN: 2575-1883 (Print); ISSN: 2575-1891 (Online)

# A Hybrid Classification Model of Artificial Neural Network and Non Linear Kernel Support Vector Machine

**Lena Anyango Onyango*, Anthony Gichuhi Waititu, Thomas Mageto, Mutua Kilai**

Department of Statistics and Actuarial Sciences, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya

## Email address:

anyangoonyangolena@gmail.com (L. A. Onyango), agwaititu@gmail.com (A. G. Waititu), ttmageto@jkuat.ac.ke (T. Mageto), kilaimutua@gmail.com (M. Kilai)

*Corresponding author

## To cite this article:

Lena Anyango Onyango, Anthony Gichuhi Waititu, Thomas Mageto, Mutua Kilai. A Hybrid Classification Model of Artificial Neural Network and Non Linear Kernel Support Vector Machine. *International Journal of Data Science and Analysis.* Vol. 8, No. 2, 2022, pp. 47-58. doi: 10.11648/j.ijdsa.20220802.15

**Received**: March 8, 2022; **Accepted**: March 28, 2022; **Published**: April 22, 2022

**Abstract:** Machine Learning Algorithms are employed in characterization, pattern recognition, and prediction. A hybrid model helps in reducing the computational complexity, improves accuracy, and results in an effective method for classification. The misclassification of the individual classifier is often excluded in a hybrid classifier. The objective of this research was to develop a hybrid classification model of Artificial Neural Network and non-linear kernel Support Vector Machine as an intelligent tool for achieving better classification performance and minimizing error rates. This study further evaluated the irreducibility and identifiability statistical properties of the ANN-SVM model. To achieve the hybridization of ANN and SVM, the research first obtained weights from the fitted Support Vector Machine model, and these weights were used as the initial weights in the Artificial Neural Network structure. The experiment was carried out in three distinct phases: selection of input features using the Boruta Wrapper Algorithm, classifier learning, and classifier combined effect and classification optimization. The study findings suggest that the hybrid ANN-SVM approach gives a higher performance accuracy of 89.7% and is more precise as compared to single ANN, SVM data mining algorithms. Therefore, the hybrid of ANN-SVM is the best binary classification system for classifying diabetes mellitus. The statistical software used for analysis was R.

**Keywords:** Hybrid, Artificial Neural Network (ANN), Support Vector Machine (SVM), Classification

## 1. Introduction

Machine Learning (ML) is an example of a computational method that improves performance by automatically learning from experience and making more accurate predictions. Machine Learning techniques are now being used in a variety of fields including education, healthcare, business and also recommendation systems among others. Machine Learning Algorithms are used in healthcare to detect interesting patterns for disease diagnosis and treatment. Classification is an essential technique for analyzing big data and several models for predicting classification accuracy exist including Artificial Neural Network, Support Vector Machine and Discriminant Analysis. SVM is a supervised learning method developed by Vapnik et al. (1995), which he later modified in (1998). It is useful for statistical classification and regression analysis. SVM models are designed to find the hyper plane that has largest margin between the target classes and categorize them easily. The use of a SVM algorithm to solve pattern recognition and classification tasks is a novel statistical learning-based technique. Support Vector Machines in specific, have been developed to tackle classification problems involving the supervised learning concept. The functionality of the human brain is mimicked by an Artificial Neural Network (ANN). It is frequently depicted as a node network known as neurons that are artificial. All of those nodes are capable of communicating with one another. Neurons are frequently depicted by a notation (0 or 1) and that each node can be designated a weight that describes its power or significance in the system. The basic ANN method weights are adjusted based on the defect within network outputs and the actual output, attempting to minimize the total error. According to Diabetes Atlas, diabetes affects

approximately 194 million individuals globally, with that figure likely to increase to 333 million by 2025. Diabetes type 2 is spreading rapidly, accounting for approximately 85 percent to 95 percent of all diabetes in developed countries and even higher in developing countries.

# 2. Literature Review

This section discusses the reviewed literature on Support Vector Machine and Artificial Neural Network. A medical diagnosis is a classification problem [1]. F. O. D, A. F. A and A. A. concluded that correct cancer diagnosis is critical in order to save human lives [2]. On procured internet cancer datasets, he compared the performance of the ANN classifier and SVM classifiers. T. T. K. M. M. Jahangir and X. S. proposed a hybrid approach for maximizing connections and criminal classification [3]. A. Syahid, S. Ali and S. Roselina, suggested a methodology that combined Artificial Neural Networks with algorithms for Artificial Bee Colonies [4]. The ABC algorithm is then used to counter the disadvantage of ANN since it converges on local optima. M. Sung-Hwan, L. Jumin and H. Ingoo, demonstrated a method for predicting bankruptcy using hybridized genetic algorithms and Support Vector Machines [5]. C. Tanujit,. C. Swarup and C. Ashis Kumar, suggested a new hybrid model for analyzing business school data based on Classification Tree (CT) as well as Artificial Neural Network (ANN) [6]. A comparison of different supervised models with the new proposed model utilizing various performance metrics is presented. The results demonstrate that the proposed blended CT-ANN model outperforms the existing supervised learning model in predicting student placing. To improve accuracy, C. Panayiotis et al. [7] present a new hybrid prediction model that combines Fuzzy Cognitive Maps (FCM) and Support Vector Machines (SVM). The experimental results show that the proposed model outperformed SVM model as well as other commonly used supervised machine-learning methodologies such as weighted k-NN, Linear Discriminant Analysis and Decision Trees. C. R. Abrunhosa,. P. Leonardo Antonio Monteiro, B. Laura, B. d. B. and R. Am'alia Faria dos, investigates the ability of Neural Network (ANN) as well as Support Vector Machine (SVM) modeling techniques to distinguish between patients with chronic heart failure who have a higher or lower likelihood of dying while hospitalized [8]. The SVM model outperforms the others based on the computational findings. Y. B, W. YT, Y. JB and W. JY, employed two techniques: Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) for the accident duration prediction, the following are the findings: both the ANN and SVM models were useful in predicting traffic accident period within reasonable limits [9]. For long-duration incident cases, the ANN model performs better. The SVM model outperforms the ANN model in terms of overall performance for predicting traffic accident duration. A comparative study has been shown by S. Amit Kumar et al. [10] where for junk mail identification, the efficiency of Naive Bayesian is contrasted to that of ANN. The Neural Network method was found to be more reliable and accurate than the Naive Bayesian method in this study.

To predict bus arrival times, Y. Bin et al. [11] demonstrated a hybrid model relying on the Support Vector Machine (SVM) as well as Kalman Filtering technique. The results demonstrate that the proposed hybrid model is viable and appropriate in the field of bus arrival time forecasting and that it outperforms the Artificial Neural Network in general. Many researchers proposed a new hybrid technique involving an evolutionary methodology to enhance the classifier performance. ANN has a number of drawbacks including lengthy training times, rising computational costs, convergence at local optimization and weight modification. To compensate for these shortcomings, Neural Networks could be merged with some other techniques.

# 3. Methodology

### *3.1. Summary*

As classification algorithms, we discussed the Artificial Neural Network and the Support Vector Machine. Lastly, we considered Hybrid of ANN and SVM as a measure to increase the performance of the two models.

Generally a statistical model is given by:

$$Y = M(x) + \varepsilon \tag{1}$$

$Y$ is the dependent variable, $Y$ represents the output for classification, $x_1, x_2, x_3 \dots x_k$ are the regressors or independent variables, $\varepsilon$ is the error term. The mean of the dependent variable, $E(\frac{Y}{x})$ is expressed in a function $m(x)$ of the dependent variable. $M(x)$ is the non-linear function relating $E(Y)$ to the independent variable $x$.

$$E[\varepsilon] = 0; Var[\varepsilon] = \sigma^2 v \tag{2}$$

Equation (2) is the assumptions of the error term.

For the purpose of classification, the mean function can be estimated by various classification models e.g. Artificial Neural Networks, Support Vector Machine or even the Hybrid to enhance the overall performance of the two.

### *3.2. Artificial Neural Network*

An Artificial Neural Network (ANN) is a highly connected collection of primary processors also known as neurons. ANN is a data-dependent, non-parametric technique. ANN are robust functions and analytic techniques for predicting and classifying problems that can model very complex nonlinear functions to high accuracy levels using a learning process similar to the learning process of the human brain cognitive system. As a learning algorithm, ANN uses the technique of Back Propagation (BP). The BP methodology is supervised, which means that it layout process inputs to expected output by reduction of errors between preferred and calculated outputs based on inputs and the network learning. By minimizing sum of squared errors, the method achieves the learning process. A supervised learning model provides

the network with a set of input vectors as well as the desired output of the network. A typical ANN has one input layer, one output layer and one or more hidden layers. Thus every layer has a large number of nodes and neurons in each layer have been linked to neurons in subsequent layers via varying network weights. Every single node in the network is represented by a neuron. Input feature vectors are given to the input-layer neurons. Weights of neuron connectivity are multiplied by the signals received by each neurons in the output and hidden layers. The neuron then produces output by combining the signals and passing them through a transfer function.

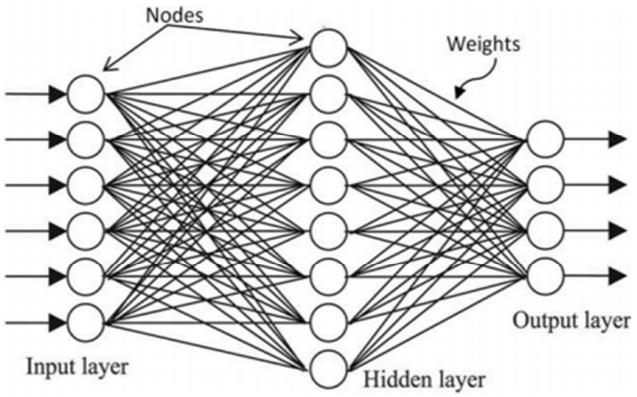


***Figure 1.*** *ANN Structure.*

The activation or transfer function determines the neuron's output in response to a specific input. Neural Networks support activation functions such as step functions, linear functions, and sigmoid (i.e., logistic function and hyperbolic tangent function) [12].

Weights $W_{hj}\ for\ h \in \{1, \dots, H\}\ and\ j \in \{0, \dots, d\}$ are used to connect the nodes in the input and hidden layers in ANN. Weights $\alpha_h$ connect the hidden output layers for $h \in \{0, \dots, H\}$. Taking into account an input vector $x = (x_1, x_2, x_3 \dots x_d) \in R^d$, then the input $v_h(x)$ to the $h^{th}$ hidden node is the value

$$v_h(x;\theta) = W_{h0} + \sum_{j=1}^{d} W_{hj} x_j \tag{3}$$

The output $\emptyset_h(x;\theta)$ of the $h^{th}$ hidden node is the value

$$\emptyset_h(x;\theta) = \psi(v_h(x;\theta)) \tag{4}$$

The value represents the net input to the node of output

$$O_H(x;\theta) = \alpha_0 + \sum_{j=1}^{H} \alpha_h \emptyset_h(x,\theta) \tag{5}$$

Finally, the net's output $M(x;\theta)$ is the value

$$M(x;\theta) = \psi\left(O_H(x;\theta)\right) \tag{6}$$

All parameters are denoted by $\theta$. $(\alpha_0, \dots, \alpha_H)$ and $W_{hj}, h = 1, \dots, H, j = 0, \dots, d$, of the said network. In addition, we compose $\alpha = (\alpha_{O,\dots,}\alpha_H)^T$ and $W = (W_{hj}, h = 1, \dots, H, j = 0, \dots, d)$.

### *3.2.1. The Bipolar or Hyperbolic Tangent Activation Function*

This is represented as:

$$\psi(x) = 2\left\{\frac{1}{1+\exp(-\alpha x)}\right\} - 1 = \tanh(\frac{\alpha x}{2}) \tag{7}$$

Noise is easily handled by Neural Networks and correlations between dependent and independent variables are simply identified. Despite this, it has issues with over fitting as well as local minima. Furthermore, computation takes a lot of time and it is also difficult to interpret in large networks [13].

As mentioned before, the classic BP algorithm employs a gradient descent method that aims to reduce the difference in mean square error between computed and expected ANN network outputs. Assuming n process inputs, x, and m desired outputs, d, and a network containing a sample size of $N: \{X_{it}, d_{jt} | i = 1,2, \dots, n; j = 1,2, \dots, m; t = 1,2, \dots, N\}$, hence the network outputs' average square error is:

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{m}(d_{jt} - y_{jt})^2 \tag{8}$$

where $y_{jt}$ denotes the computed output.

Each and every network connection weights is adjusted as below using the gradient descent method:

$$\Delta W = -\eta \Delta E(W) = -\eta \frac{\partial E}{\partial W} = \sum_{t=1}^{1}\sum_{j=1}^{m}(d_{jt} - y_{jt})\frac{\partial y_{jt}}{\partial W} \tag{9}$$

$\eta$ is a very small number, like 0.1, that represents the algorithm's rate of learning during the training.

### *3.2.2. Back Propagation (BP)*

Back Propagation is a type of gradient descent method that works on a coordinate level. The weights of a unipolar $\psi(x)$, are adjusted as follows:

$$W^{r+1} = W^r + \Delta W$$
$$\alpha^{r+1} = \alpha^r + \Delta\alpha \tag{10}$$

weights for individuals, we get $r^{th}$ iteration weights given below:

$$\alpha_h{}^{(r+1)} = \alpha_h{}^{(r)} - \gtrdot_1 \left\{\frac{\partial S(Y,X;\theta^{(r)})}{\partial \alpha_h}\right\} for\ i = 1, \dots, n\ and\ h = 1, \dots, H \tag{11}$$

Similarly,

$$W_{hj}{}^{(r+1)} = W_{hj}{}^{(r)} - \gtrdot_2 \left\{\frac{\partial S(Y,X;\theta^{(r)})}{\partial W_{hj}}\right\} for\ i = 1, \dots, n, h = 1, \dots, H\ and\ j = 0, \dots, d \tag{12}$$

$\gtrdot_1$ and $\gtrdot_2$ denote the gain in steps.

The weights are changed up to when the stopping criterion

is reached. Every weight is modified $n$ times during every iteration using this method. This means that every weight is modified $I_n$ times for $I$ iterations. As a result, the technique is very slow, since $I$ is usually big. The methodology is certainly not very stable, resulting in estimates that are asymptotically inefficient.

### *3.3. Support Vector Machine*

V. Vladimir, was the first to introduce the Support Vector Machine (SVM) [14]. SVM is a classification methodology used on both non-linear data and linear data. SVM is utilized to obtain the perfect categorization function in training data to distinguish objects of two categories. SVM fixes the binary class (linear separable) issue by determining the best (maximum) marginal hyper plane (MMH) that divides the classes. Support vectors and margins are used to find this hyper plane [15]. A nonlinear mapping is used in the transformation of nonlinear data into a high-dimensional space. It seeks the linear optimal separating hyper plane within the new dimension [16].

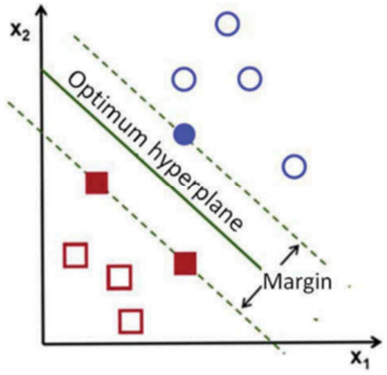


***Figure 2.*** *SVM Structure.*

As previously stated, the main principle of SVM is to find the best separation plane under linearly separable conditions. Figure 2 [17] depicts the basic idea by depicting the categorization of a set of data with two distinct classes of data, category I (circles) and category II (squares). The SVM seeks the best hyper plane (linear boundary) between the two classes and aligns itself to optimize the margin. The statistics point closest to the partitioning boundary which are used to describe the margin are referred to as support vectors. Assume in a specific training set of samples, $G = \{x_i, y_j\}_1^N$ in which case for each and every input variable $x_i \in R^d$, there is indeed a desired output in the class defined by $\in \{+1, -1\}$, $y_i$ can be given as +1 or -1, showing the category that the point $x_i$ features. $x_i$ is a real-vector in d dimensions. The classification function of the form is created using SVM:

$$M(x) = \big(w, \Phi(x)\big) + b, \Phi: R^d \to F, w \in F \tag{13}$$

The data is represented by $\Phi(x)_1^N$ in the feature space, $b$ and $\{w_i\}_1^N$, are the coefficients.

These are calculated by trying to minimize the risk function in the following way:

$$R(C) = C\frac{1}{l}\sum_{i-1}^{N} L_e\big(y_i, f(x)\big) + \frac{1}{2}||w||^{\widehat{2}} \tag{14}$$

In the above, C is a regularization constant, and $L_e(y_i, f(x))$ the error function that evaluates the estimated differences between the desired output $y_i$ and the computed output $f(x_{i)}$.

$\frac{1}{2}\big||w|\big|^{\widehat{2}}$ determines the trade-off between training error as well as generalization ability. The second term given by equation (14) is a measure of how flat the function is. Equation (14) becomes the following restricted function when the relaxation factor $\xi, \xi^*$ is introduced:

$$minimiseJ = \frac{1}{2}\big||w|\big|^{\widehat{2}} + C\sum_{i=1}^{N}(\xi_i^* + \xi_i) \tag{15}$$

$$subject\ to \begin{cases} y_i - [w, \Phi(x)] - b \le \varepsilon + \xi_i^* \\ [w, \Phi(x)] + b \le \varepsilon + \xi_i^* \\ \xi_i^*, \xi_i \ge 0 \end{cases} \tag{16}$$

Finally, the equation above is expressed in the explicit form by applying Lagrange multipliers and the use of the best constraints.

$$f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x_i, x) + b \tag{17}$$

In equation (17), $\alpha_i^*$ and $\alpha_i$ are Lagrange multipliers that satisfy the equalities. $\alpha_i^* * \alpha_i = 0, \alpha_i^* \ge 0\ and\ \alpha_i \ge 0$. Where $i$ represents a range of integer values from 1 to N. $\alpha_i^*$ and $\alpha_i$ are obtained by minimizing the equation to obtain:

$$\min J = \frac{1}{2}\sum_{i=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_i^* - \alpha_i)\,K(x_i, x_k) + \sum_{i=1}^{N}\alpha_i^*(y_i - \varepsilon) + \sum_{i=1}^{N}\alpha_i^*(y_i + \varepsilon)$$

$$subject\ to \begin{cases} \sum_{i=1}^{N}\alpha_i = \sum_{i=1}^{N}\alpha_i^* \\ 0 \le \alpha_i \le C \\ 0 \le \alpha_i^* \le C \end{cases} \tag{18}$$

Despite the fact that function that is nonlinear $\Phi$, all $\Phi$ related computations may be reduced to its most basic form $\Phi(x)^T.\Phi(y)$, which can be substituted for the kernel function $K(x, y) = \Phi(x).\Phi(y)$. The kernel function has the benefit of allowing you to work with vector space of any dimension without needing to explicitly compute the map. Kernel functions serve as a link between linear and nonlinear methodologies usually stated in relation to dot product. Functions of the kernel are employed in a non-linear mapping to translate the input data to a higher dimensional space. Linear, Sigmoid, Radial Basis Function (RBF) and Polynomial kernels are the four most popular SVM kernel

function types. They're as follows:

$$Linear: K(x_i, x_j) = x_i^T x_j$$

$$Sigmoid: \tanh(\gamma x_i^T x_j + r)$$

$$Radial\ basis\ function\ (RBF): K(x_i, x_j) = e^{-(\sigma x_i - x_j^2)}, \sigma \geq 0$$

$$Polynomial: K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \quad (19)$$

*Radial Basis Function*

As the classification model, the SVM's Radial Basis Function (RBF) kernel is utilized, because the RBF kernel function can analyze higher-dimensional data. The kernel's output is determined by $x_j$'s Euclidean distance from $x_i$ (one of these will be the support vector and the other will be the testing data point). The center of the RBF will be the support vector, which will establish the zone of control this support vector has on the data space. The RBF kernel function is defined as follows:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0 \quad (20)$$

where $\gamma$ is a kernel variable and represents the training vector. A higher $\gamma$ value results in a smoother decision boundary and more frequent decision boundary. This is due to the fact that a RBF with a large $\gamma$ allows a support vector to exert a strong influence over a wider area. The classifier is obtained after applying the best optimal parameters to the training dataset. To determine generalization accuracy, the constructed classifier is utilized to classify the testing dataset.

### *3.4. Proposed Framework*

The proposed method aimed to develop a diabetes design at a preliminary phase using an Artificial Neural Network and a Support Vector Machine. SVM and ANN machine learning algorithms have different structures for learning from a dataset. The workflow structure of the proposed approach is represented in Figure 3.

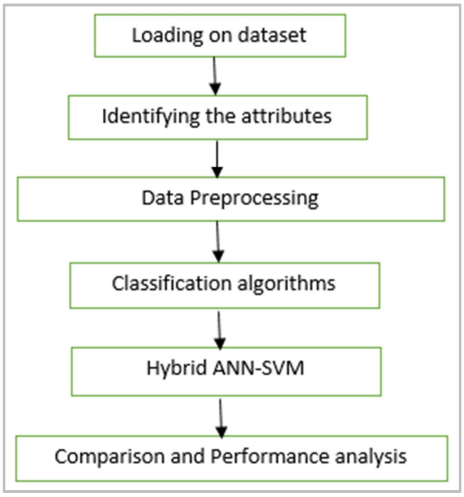


***Figure 3.*** *Proposed Framework.*

*Model Development*

Any project involving data mining goes through three distinct phases in order to construct a model: creating the learning data set, constructing a model from the data available and verifying it. Any model differs from the others depending on which technique is used at each phase. Diabetes mellitus illness diagnosis is described as a classification task, the study then illustrates the strategies that may be employed at every stage to develop a reliable diagnosis model.

#### *3.4.1. Data Preprocessing*

The first step in developing a Machine Learning Technique is pre-processing. It converts raw data into meaningful format. In general, real-world data is insufficient, inconsistent and rife with errors. Preprocessing is performed on the given data to reduce errors. Before one can use the classification model, one must first process the data. The most common pre-processing steps are Feature Selection and Normalization.

#### *3.4.2. Feature Selection Algorithm*

Most of the time, the majority of the variables are insignificant to the supervised machine learning algorithm. Due to the fact that using raw set of data directly may impede the whole learning process, the dimensions are estimated. The dataset's features was chosen using the Boruta Wrapper Technique that gives unbiased choice of significant features. The study chose the features that comprise of data that are highly correlated. This phase is carried out using the feature selection algorithm, which may be carried out using the "manual process" or the Boruta Wrapper method. The Boruta package gives a consistent and unbiased choice of significant attributes from a data system, on the contrary the manual method is prone to errors.

#### *3.4.3. Normalization*

For a number of reasons, most data mining approaches necessitate data normalization. Data normalization is used to smooth data in order to improve data generalization and performance. Normalization in classification algorithms speeds up the learning process and avoids high-range attributes from dominating other features, as in distance-based approaches [18]. Another advantage of this method is to avoid numerical difficulties during calculation.

The maximum and minimum values are utilized to define the function normalization. The normalization formula is given as:

$$X_{new} = \frac{X_t - X_{min}}{X_{max} - X_{min}}(D_{max} - D_{min}) + D_{min} \quad (21)$$

where;

$X_t$ is the normalized value

$X_{min}$ minimum value for the statistic variable

$X_{max}$ maximum value for the statistic variable

$D_{min}$ and $D_{max}$ are maximum and minimum values required for normalization

### *3.5. Performance Metrics*

Several measurements, for example, accuracy (ACC), sensitivity, precision, Receiver Operating Characteristic (ROC) curve and specificity can be utilized to assess the effectiveness of a classification model. Before diving into each of these metrics, it's important to understand the following terminologies: TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative). These terms which are considered as a result of a classifier are most effectively represented by a confusion matrix (Table 1).

***Table 1.*** *Confusion Matrix.*

| | **Actual Class** | | |
|---|---|---|---|
| | | **Negative** | **Positive** |
| **Predicted Class** | Negative | TN | FN |
| | Positive | FP | TP |

### *3.6. Statistical Properties*

#### *3.6.1. Model Irreducibility*

Remember a neural network's definition given by equations (3)–(6), under which we primarily take the unipolar activation function $\psi(x)$ for example, in this study because it has values ranging from [0,1]. To make it easier to refer to the literature that exist, we take into consideration the case of the function of bipolar activation given by $\tilde{\psi}(u)$ for the neurons that are hidden as well as the output neuron identity in the following two sections. The network is now presented by:

$$v_h(x;\theta) = W_{h0} + \sum_{j=1}^{d} W_{hj}x_j \tag{22}$$

The $h^{th}$ hidden node's output $\emptyset_h(x;\theta)$ is the value

$$\tilde{\emptyset}_h(x;\theta) = \tilde{\psi}(v_h(x;\theta)) \tag{23}$$

The value

$$\tilde{M}(x;\theta) = \tilde{O}_H(x;\theta) = \tilde{\alpha}_0 + \sum_{h=1}^{H} \tilde{\alpha}_h \tilde{\emptyset}_h(x,\theta) \tag{24}$$

is the cumulative input to the output node.

We talk about identifiability, or how the mapping $(x_{1,\dots,}x_d) \longrightarrow \tilde{M}(x;\theta)$ respectively $\longrightarrow M(x;\theta)$ determines the parameters up to a certain point.

We can immediately deduce

$$\tilde{\psi}(u) = 2\psi(u) - 1, \psi(u) = \frac{1}{2}\left(1 + \tilde{\psi}(u)\right) \tag{25}$$

from the definitions of $\emptyset$ and $\tilde{\emptyset}$.

Therefore, we have:

$$\tilde{O}_H(x;\tilde{\theta}) = \tilde{\alpha}_0 + \sum_{h=1}^{H} \tilde{\alpha}_h\left(2\psi\left(v_h(x;\theta)\right) - 1\right) = (\tilde{\alpha}_0 - \sum_{h=1}^{H} \tilde{\alpha}_h) + \sum_{h=1}^{H}(2\tilde{\alpha}_h)\,\psi\left(v_h(x;\theta)\right) = O_H(x;\theta) \tag{26}$$

If we relate $\tilde{\theta}$, $\theta$ by

$$\alpha_0 = \tilde{\alpha}_0 - \sum_{h=1}^{H} \tilde{a}_h,\ a_h = 2\tilde{a}_h, h = 1,\dots,H \tag{27}$$

and both parameter vectors $\tilde{\theta}$, $\theta$ have the same $W_{hj}$. As a result, if the parameter vector from the mapping is identifiable, that is to say, it is identifiable for $O_H(x;\theta)$ if and only if it is identifiable for $\tilde{O}_H(x;\theta)$. Finally, we note that $\psi(u)$, $M(x;\theta)$ and $O_H(x;\theta)$ are uniquely determined by their continuity and strict monotonicity. As a result, we can investigate identifiability issues for $\tilde{M}(x;\theta)$ rather than $M(x;\theta)$. We then write for the activation function that is bipolar and $\theta$ as well as the relevant variable to simplify notation. Before delving into model irreducibility, we need to first describe redundancy. If at all there is another network with lesser neurons that otherwise gives the same input-output map as a neural network with a given $\theta$, the neural network with that $\theta$ is redundant. If one of the following conditions is met, a net with $\psi(x)$ is reducible.

(a) $\alpha_i = 0$ in some case $i = 1,\dots,H$.

(b) One of the functions, $v_i(x,\theta)$ is a fixed value

(c) There are two distinct indexes $i_1, i_2 \in \{1,\dots,H\}$ for which the functions $v_{i_1}(x)$ and $v_{i_2}(x)$ are key indicator equivalent.

That is to say $v_{i_1}(x;\theta) = \pm v_{i_2}(x;\theta)$.

A neural network that fulfills any of the three criteria listed above is redundant because the input-output function may be performed by yet one more network containing lesser hidden neurons. This is accomplished by removing one neuron. We notice that if at all condition (a) holds, the *i*-th hidden node contributes nothing to net input $M(x;\theta)$. If we delete the *i*-th node, the input-output map remains unchanged. If and only if a net is reducible due to (b) is true, then $v_i$=c, where c is said to be a constant. As a result, the *i*-th node can be removed and $\alpha_0$ replaced with $\alpha_0 + \alpha_{0\,i}\psi(c)$. This condition is only possible if, for a given fixed *i* $W_{ij} = 0$ for all $j = 1,\dots,d$. So, $v_i = W_{io}$ in this case. Finally, if a net can be reduced due to (c), we are able to write $v_{i_1}(x;\theta) = \tau v_{i_2}(x;\theta)$ where $\tau = 1$ or $\tau = -1$. The nodes $i_1$ and $i_2$ then contribute a total value of

$$\alpha_{i_1}\psi\left(v_{i_1}(x;\theta)\right) + \alpha_{i_2}\psi\left(v_{i_2}(x;\theta)\right) = \alpha_{i_1}\psi\left(\tau v_{i_2}(x;\theta)\right) + \alpha_{i_2}\psi\left(v_{i_2}(x;\theta)\right)$$

$$= \tau\alpha_{i_1}\psi\left(v_{i_2}(x;\theta)\right) + \alpha_{i_2}\psi\left(v_{i_2}(x;\theta)\right) = (\tau\alpha_{i_1} + \alpha_{i_2})v_{i_2}(x;\theta) \tag{28}$$

to $M(x;\theta)$.

Since $\psi(x)$ is an odd function, this relationship exists, that is, $\psi(\tau x) = \tau\psi(x)$.

We can then remove node $i_1$ and substitute $\alpha_{i_2}$ with $\tau\alpha_{i_1+}\alpha_{i_2}$ in this type of reducibility. The existence of insignificant hidden layer's neurons is responsible for conditions (a) and (b). Schwarz Information Criterion (SIC) is used as our model selection method to control these conditions, which is denoted as:

$$SIC(h) = \ln(\hat{\sigma}^2) + (h(2+d)+1)\,\frac{\ln(n)}{n} \tag{29}$$

The first term is a measure of goodness-of-fit, and the middle term a penalty for complexity. We begin putting a single hidden neuron then use the SIC criterion to determine SIC (1). The SIC (2) is then determined after the addition of a second hidden neuron. The procedure is repeated until an additional hidden neuron doesn't really ameliorate the SIC. We hence estimate *h+1* models so that we can select a model containing *h* neurons.

This methodology make certain that $\alpha_i \neq 0 \forall i$ and $W_{ij} \neq 0$ for $j = 1, \dots, d \ \forall$i. To guarantee that $\theta$ is irreducible and therefore non-redundant, we simply need to make the following assumption. There are no two distinct indexes $i_1, i_2 \in \{1, \dots, H\}$ for which the functions $v_{i_1}(x; \theta)$ and $v_{i_2}(x; \theta)$ are strong indication equivalent. This assumption eliminates the irreducibility generated by condition (c). By using equation (27) and the discussion earlier in this section, the result is directly appropriate in the context of a unipolar activation function. Despite the fact that $\theta$ is now irreducible, it's indeed unidentifiable, as explored below.

#### *3.6.2. Model Identifiability*

The inability of the parameters to be identified is a fundamental issue with the ANN. As a result, we have distinct groups of variables, but the (Y, X) distributions are the same. As a result, the parameters are not one-of-a-kind. To make this clear, we denote all of the weights as follows:

$$\alpha_0 \, and \, \beta_i = (\alpha_i, W_i) \, for \, i, \dots, H \, where \; W_i = (W_{i0}, W_{i1}, \dots, W_{id}) \tag{30}$$

The sources of unidentifiability are now discussed. Each ANN is unidentifiable. The two transformations listed below have no effect on the input-output map of a neural network:

i. The possible combination of $\beta_i$'s is if we swap two hidden nodes, assume $h_s$ and $h_t$, where *s* and *t* signify the actual position of the node and rename them $h_t$ and $h_s$ and also rename the associated weights $\alpha_t$ and $\alpha_s$ as well as $W_t$ and $W_s$, $M(x; \theta)$ remains constant. This transformation solely produces H! distinct models of the same input output map.
ii. The symmetry of $\psi(x)$ causes the other invariant transformation. That is to say $\psi(x) = -\psi(-x)$. This implies therefore that if we select a hidden node $h_t$ as well as negate both $W_t$ and $\alpha_t$, the input-output map stays consistent. In practice, this indicates that $(\alpha_0, \beta_1, \dots, \beta_i, \dots, \beta_H)$ as well as $(\alpha_0, \beta_1, \dots, -\beta_i, \dots, \beta_H)$ have duplicate input-output map. This transformation on its own produces $2^H$ distinct models with identical input-output map.

In particular, they demonstrate that $(\alpha_0, \beta_1, \dots, \beta_i, \dots, \beta_H)$ and $(\alpha_0 + \alpha_i, \beta_1, \dots, -\beta_i, \dots, \beta_H)$ contain duplicate input-output map. These two transformations produce the $2^H H!$ element family. All of these transformations are referred to as $\eta$. Each of these transformations is described as being $(\eta_1, \dots, \eta_H)$ function that is composite where

$$\eta_1([(\alpha_0, \beta_1, \dots, \beta_i, \dots, \beta_H)]) = (\alpha_0, -\beta_1, \dots, \beta_i, \dots, \beta_H) \, and \, \eta_i([(\alpha_0, \beta_1, \dots, \beta_i, \dots, \beta_H)])$$
$$= (\alpha_0, \beta_i, \beta_2, \dots, \beta_{i-1}, \beta_1, \beta_{i+1}, \dots, \beta_H) \, for \, i = 2, \dots, H \tag{31}$$

# 4. Results and Discussion

There are 2818 instances in total that were grouped into two categories: diabetic and non-diabetic with six distinct risk factors: age, plasma glucose concentration of two hours in an oral glucose tolerance test, diastolic blood pressure, body mass index, two-hour serum insulin and diabetes pedigree function.

***Table 2.*** *Data Description.*

| Attributes | Description | Format | Range | Value |
|---|---|---|---|---|
| BMI | Body Mass Index | numeric | 18-47 | Kg/m^2 |
| Insu | Insulin | numeric | 0-846 | mu U/ml |
| Age | Age | numeric | 21-67 | Years |
| Bp | Blood Pressure | numeric | 0-122 | mmHg |
| Diaped | Diabetes Pedigree Function | numeric | 0.078-2.42 | - |
| Glucose | Plasma Glucose Concentration | numeric | 0-200 | mg/dl |
| Outcome | Diabetic; not Diabetic | nominal | - | 1,0 |

### *4.1. Feature Selection*

The automatic choice of features in the data that are most meaningful for use in model construction is known as Feature Selection. Most of the time, the majority of the features are insignificant to the machine learning supervised categorization. So feature selection is an important step to consider before classification. The dataset's features were chosen using the Boruta Wrapper Algorithm that gives unbiased selection of significant variables.

***Table 3.*** *Boruta Wrapper Algorithm.*

| Results from Boruta Wrapper Algorithm |
|---|
| Boruta performed 10 iterations in 12.98696 sec |
| 6 attributes confirmed important: Age, Blood pressure, BMI |
| Diabetes Pedigree, Glucose and Insulin |
| No attribute deemed unimportant |

*Boruta Wrapper Algorithm*

We chose the characteristics that contained strongly correlated data. The Boruta Wrapper was applied to a set of data that contains all of the features, where it returned all of the features as significant.

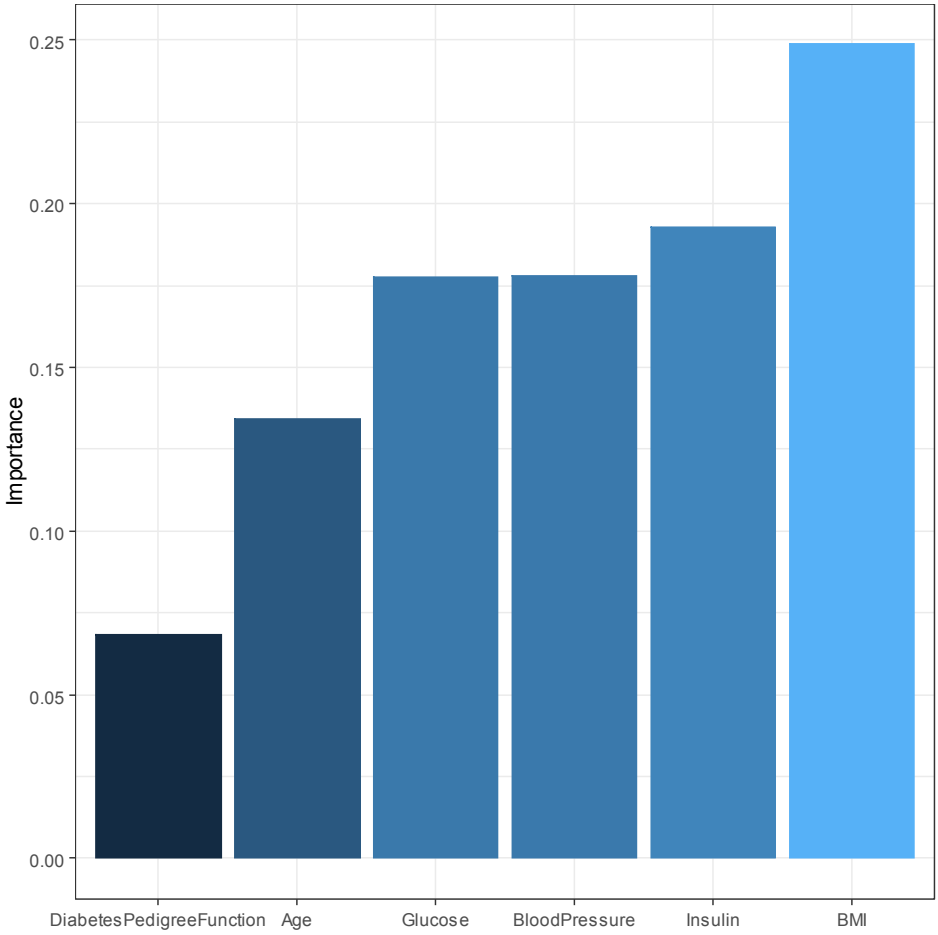


***Figure 4.*** *Variable Importance.*

### *4.2. Variable Importance*

The most significant variables are listed in descending order by a mean reduction on the variable importance plot. The top variable contributes more to the model than the bottom ones and it also has a strong predictive potential for classifying diabetics from non-diabetics.

### *4.3. Fitting SVM Model to the Data*

The SVM classifier was tested with different kernel functions. This study employed the training time and accuracy rate as a criterion to find the optimal kernel function.

***Table 4.*** *Comparison of Classification Accuracy and Training Time of SVM Classifier with Different Kernels.*

| Kernel Function | Training Time(s) | Accuracy (%) |
|---|---|---|
| Radial | 0.71474 | 85.46 |
| Polynomial | 1.630221 | 76.667 |
| Sigmoid | 1.936781 | 71.4815 |
| Linear | 1.31639 | 50.1055 |

The study used cross validation on every combination of $C \in \{0.01,0.1,1,10\}$ and $\gamma \in \{0.01,0.1,1,10\}$, since using grid search on an exponential scale expanding series of C and $\gamma$ frequently results in strong classification results. The combinations that gave the best accuracy was checked.

The C- regularization parameter governs the tradeoff between SVM uncertainty and classification error. $\gamma$ is the kernel parameter that sets the kernel width.

***Table 5.*** *SVM's Fine-tune Parameters.*

| Parameters | | Accuracy |
|---|---|---|
| C | $\gamma$ | |
| 0.01 | 0.001 | 0.035 |
| 0.01 | 0.01 | 67.048 |
| 0.01 | 0.1 | 78.353 |
| 0.01 | 1 | 77.549 |
| 0.1 | 0.001 | 65.826 |
| 0.1 | 0.01 | 78.299 |
| 0.1 | 0.1 | 78.548 |
| 0.1 | 1 | 78.626 |
| 1 | 0.001 | 78.284 |
| 1 | 0.01 | 78.482 |
| 1 | 0.1 | 78.48 |
| 1 | 0.5 | 83.33 |
| 1 | 0.75 | 78.682 |
| 1 | 1 | 78.721 |

| Parameters | | Accuracy |
|---|---|---|
| C | γ | |
| 1 | 1.25 | 78.735 |
| 1 | 1.5 | 78.734 |
| 10 | 0.001 | 78.46 |
| 10 | 0.01 | 78.527 |
| 10 | 0.1 | 78.665 |
| 10 | 1 | 78.085 |

### 4.3.1. Results from the Fitted SVM Model

*Table 6. Results from the Fitted SVM Model.*

| Parameters | Description |
|---|---|
| SVM-Type | C-classification |
| SVM-Kernel | Radial |
| Cost | 1 |
| Number of support vectors | 906 |

The performance of the classifier were analyzed using the confusion matrix technique. A confusion matrix is a table matrix that displays correct and incorrect classification.

*Table 7. Confusion Matrix of the Fitted SVM Model: out-of-sample.*

| | Reference | | |
|---|---|---|---|
| | | 0 | 1 |
| Prediction | 0 | 555 | 53 |
| | 1 | 99 | 233 |

In the study, the test data consisted of 940 observations. In reference to the confusion matrix entries, the SVM model's cumulative number of correct categorizations is (555+233) = 788 and the total number of the incorrect classification is (53+99) =152. 53 cases of diabetes were misclassified as non-diabetic while 99 cases of non-diabetes were misclassified as having diabetes. The accuracy of the model is 83.33%.

### 4.3.2. Out-of-Sample Statistics for the Fitted SVM Model

*Table 8. Out-of-sample Statistics for the Fitted SVM Model.*

| Accuracy | Sensitivity | Specificity |
|---|---|---|
| 0.8383 | 0.8147 | 0.8486 |

Specificity and Sensitivity are measurements that define exactly how good the classifier discriminates among cases with negative and positive classes, with 84.86 percent and 81.47 percent respectively.

### 4.4. Fitting ANN Model to the Data

*Table 9. Neural Network Parameters.*

| Network | Back Propagation |
|---|---|
| Learning rule | Levenberg-Marquardt |
| Transfer function | Hyperbolic Tangent |
| Learning method | Supervised |
| No. of inputs | 6 |
| No. of hidden layers | 1 |
| No. of hidden nodes | 10 |
| No. of outputs | 1 |
| Network structure | 6-10-1 |

For the classification of diabetes, a feed-forward ANN was used. The network was trained using the Levenberg-Marquardt Back-Propagation Technique, as well as gradient descent with momentum weight as well as bias learning function.

### 4.4.1. Testing Accuracy of ANN with Varying Numbers of Hidden Neurons

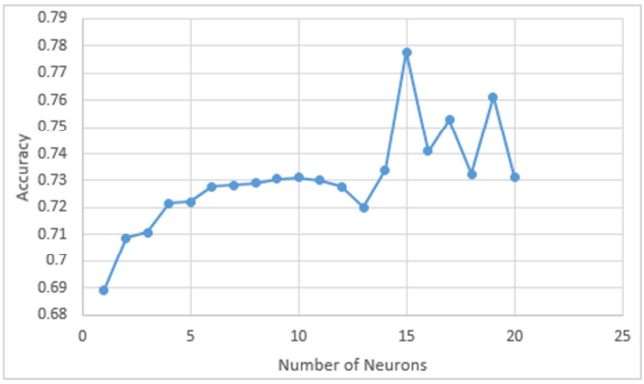


*Figure 5. Number of Neurons.*

The figure shows that the best average classification was achieved when number of hidden neurons was 10.

### 4.4.2. Results from the Fitted ANN Model

*Table 10. Confusion Matrix of the Fitted ANN Model: out-of-sample.*

| | Reference | | |
|---|---|---|---|
| | | 0 | 1 |
| Prediction | 0 | 531 | 77 |
| | 1 | 201 | 131 |

### 4.4.3. Out-of-Sample Statistics for the Fitted ANN Model

*Table 11. Out-of-sample Statistics for the Fitted ANN Model.*

| Accuracy | Sensitivity | Specificity |
|---|---|---|
| 0.7043 | 0.6298 | 0.7254 |

### 4.5. Fitting a Hybrid ANN-SVM Model

The study first obtained weights from the fitted SVM model. These weights were used as the initial weights in the Artificial Neural Network Structure. The study then developed an approach that merge the classification algorithms.

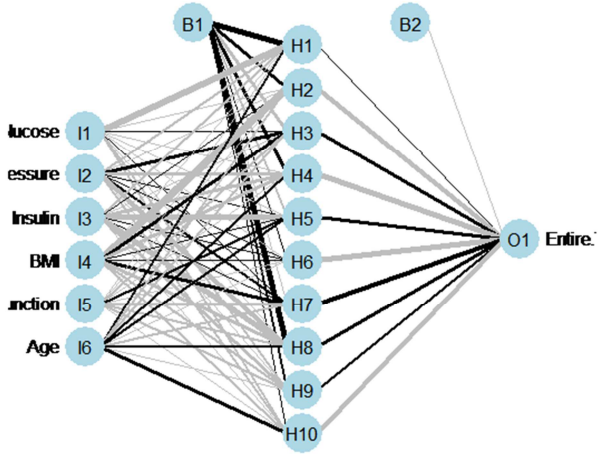


*Figure 6. ANN-SVM Hybrid plot.*

### *4.5.1. Results from the Fitted ANN-SVM Model*

***Table 12.*** *Confusion Matrix of the Fitted ANN-SVM Model: out-of-sample.*

| | Reference | | |
|---|---|---|---|
| | | **0** | **1** |
| Prediction | 0 | 590 | 27 |
| | 1 | 70 | 253 |

### *4.5.2. Out-of-Sample Statistics for the Fitted ANN-SVM Model*

***Table 13.*** *Out-of-sample Statistics for the Fitted ANN-SVM Model.*

| **Accuracy** | **Sensitivity** | **Specificity** |
|---|---|---|
| 0.8968 | 0.9036 | 0.8939 |

The problem of solving diabetes diagnosis is attributed with the decrease of number of false positive rate while maintaining a high degree of true rate of positive diagnosis i.e. sensitivity.

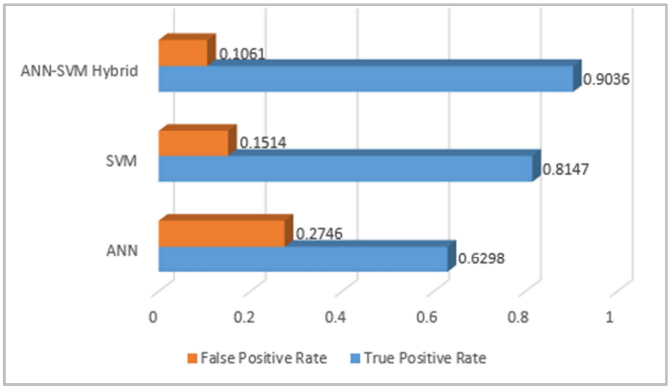


***Figure 7.*** *True Positive Rate and False Positive Rate.*

True positive has higher ratio than wrongly classified observations. Many methods have been suggested to minimize the number of false positive rate while in the same time maintaining a high sensitivity (true positive rate) and the formulated ANN-SVM model is one of them.

### *4.6. Class Imbalance Problem*

One class, for instance, fraud detection, statistics or health data may be uncommon. The term "imbalance" refers to the fact that one class is overrepresented in comparison to the other. This is a prevalent problem in real-world arena in detecting uncommon but important cases. When performing classification where we have a problem of class imbalance, sensitivity, f-measure and specificity measurement should be considered rather than accuracy. Accuracy will give bias results. Sensitivity and Specificity are measures which define how well the classification algorithm distinguish between cases of positive and negative categorization.

### *4.7. Comparing the Performance of ANN, SVM and ANN-SVM Models*

Comparative analysis was done between the machine learning classifier algorithms such as SVM, ANN and hybrid ANN-SVM based on their performance metrics.

***Table 14.*** *Performance Evaluation Metrics for Different Classifiers.*

| | **Sensitivity** | **Specificity** | **Precision** | **Error Rate** |
|---|---|---|---|---|
| ANN | 0.6298 | 0.7254 | 0.3946 | 0.2957 |
| SVM | 0.8147 | 0.8486 | 0.7018 | 0.1617 |
| ANN-SVM | 0.9036 | 0.8939 | 0.7833 | 0.1032 |

***Table 15.*** *Performance Evaluation Metrics.*

| | **f-measure** | **AUC** | **ROC** | **MCC** | **Computational Time (s)** |
|---|---|---|---|---|---|
| ANN | 0.4852 | 0.7900 | 0.6776 | 0.3085 | 9.507085 |
| SVM | 0.7540 | 0.8584 | 0.8317 | 0.6385 | 4.650424 |
| ANN-SVM | 0.8392 | 0.9017 | 0.8988 | 0.7680 | 2.076718 |

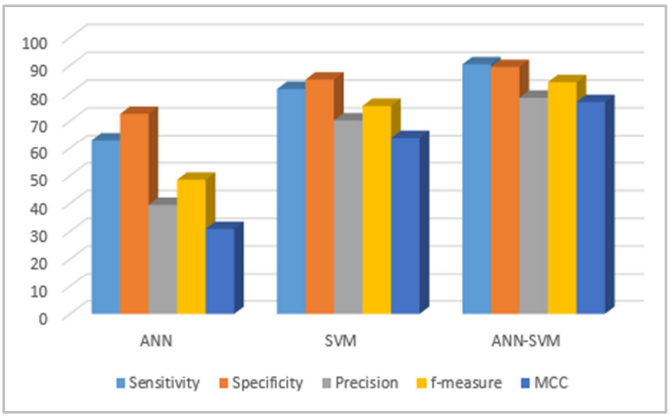


***Figure 8.*** *Comparison Analysis of Data Mining Techniques.*

After combining, it was discovered that the formulated algorithm overcomes previous drawbacks such as low classifying accuracy, computational time requirements, sensitivity and specificity of conventional ideas. The ROC curve is a useful device for assessing classifier's performance in diagnostic testing. It considers the tradeoffs between true positives and false positives rate to summarize a classifier's performance over a range.

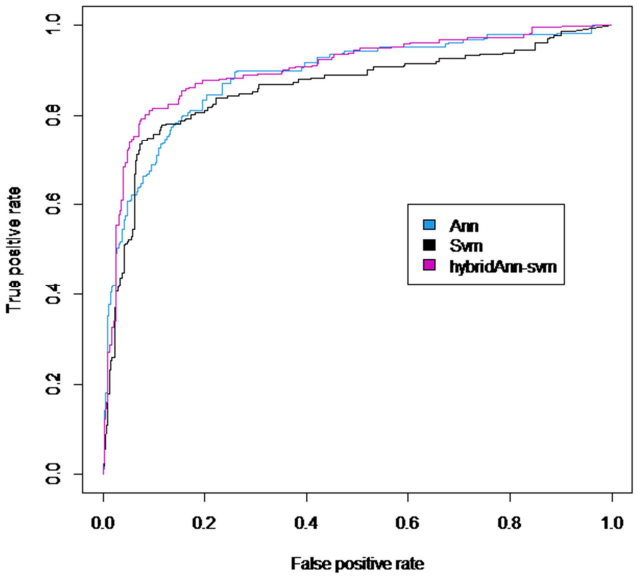


***Figure 9. AUC****-ROC curve for Hybrid ANN-SVM.*

Matthew's Correlation Coefficient (MCC) considers all four values in the confusion matrix, as well as larger values (close to 1) indicates that both classes are correctly classified, even if one category is over or underrepresented. The area under the ROC curve represents the accurateness of the model. AUC (area under the curve) varies in value from 0 to 1. A classification that is 100% has an AUC of 1. Since the AUC in this study is equal to 0.9017 it means that the model has desirable performance.

### *4.8. Discussion of Results*

To detect diabetes, the efficiency of all the three models was evaluated using parameters such as accuracy, precision, recall, specificity, f-measure, Receiver Operating Characteristic (ROC) curve, Area Under Curve (AUC) and MCC. Accuracy indicates how often the classification model is accurate in determining if a person is diabetic or not. Precision assesses a classifier's capability to make accurate positive diabetes predictions. In this study, recall or sensitivity was utilized to determine the percentage of true positive cases of diabetes identified perfectly by the classification algorithm used. Specificity refers to a classifier's ability to identify negative diabetes cases. The weighted average of precision and recall gives the f-measure, thus this score takes both into consideration. Classifiers with f-measure scores close to 1.0 are referred to as the best ones. A documented tool for determining the effectiveness of a binary classifier methodology could be the ROC curve. ROC curve is a plot of the true positive rate versus the false positive rate since the threshold for allocating observations to a specific class is varied. A classifier's area under the curve (AUC) value can range from 0.5 to 1. Values less than 0.5 showed a slew of random data that could not tell the difference between true and false. The area under the curve (AUC) of an optimal classifier is close to 1.0. If it's close to 0.5, this value is equivalent to guessing at random. It was discovered that the correctness of the Radial Basis Function kernel was 85.46%. It was ideal for classification because it had the highest accuracy when compared to other kernels. When the study carefully investigated diabetic data-set, it was discovered that it is a representation of a class that is unbalanced, with 1856 negative occurrences and 962 positive occurrences, resulting in an unbalancing proportion of 1.93. In the case of an imbalanced class, accuracy only could not give a good indication of the performance of a binary classifier. Due to the fact that it balances precision and recall, the f-measure provided better perspective into classifier performance even when the class distribution was uneven. As a result, in this case, f-measure was to be considered. The accuracy of the ANN model was 0.7043, that of the SVM model was 0.8383 and that of the ANN-SVM hybrid model was 0.8968. Recall or sensitivity, that indicates the percentage of successfully identified diabetic cases, was 0.6298 for the ANN model, 0.8147 for SVM and 0.9036 for ANN-SVM. Specificity of ANN, SVM and ANN-SVM were 0.7254, 0.8486 and 0.8939 respectively. Precision for ANN, SVM and ANN-SVM were 0.3946, 0.7018 and 0.7833 respectively. The f-measures of ANN, SVM and ANN-SVM were 0.4852, 0.7540 and 0.8392 respectively. To assess the performance of the models, we calculated the area under the curve (AUC). The AUC of the ANN model was 0.7900, while the AUC of the SVM and ANN-SVM models were 0.8584 and 0.9017 respectively. The MCC of ANN, SVM, and ANN-SVM were 0.3085, 0.6385, and 0.7680 respectively. As a result of the preceding studies, it is evidently stated that, when all parameters are considered, hybrid ANN-SVM is the optimal model to determine if a person is diabetic or not.

## 5. Conclusion and Recommendations

ANN and SVM models were fitted to the data and finally hybrid ANN-SVM model formulated by using SVM weights to initialize the Neural Network. This study explored the hybrid ANN-SVM as the finest binary classification system for classifying diabetes. Combining of the two models resulted in better performance than individual models. The Boruta Wrapper Technique was used for variable selection, which offered an unbiased choice of significant variables. All models were evaluated using various parameters such as accuracy, recall or sensitivity, specificity, precision, f-measure, ROC, MCC and AUC.

Other appropriate Feature Selection Techniques apart from Boruta Wrapper Algorithm may be explored to improve the classification rate. In investigating the performance of the models, other performance measures apart from AUC, ROC, f-measure, MCC and Computational time may be considered in future studies. Other statistical properties of the hybrid ANN-SVM model should be investigated. Apart from considering data in the medical field, future studies can use data from other fields like banking sector, insurance sector etc.

## References

[1] C. M. Amine, S. Meryem and S. Nesma, "Diagnosis of diabetes diseases using an artificial immune recognition system2 (AIRS2) with fuzzy k-nearest neighbor," Journal of medical systems, vol. 36, no. 5, pp. 2721-2729, 2012.

[2] F. O. D, A. F. A and A. A., "Classification of cancer of the lungs using SVM and ANN," Int. J. Comput. Techno, vol. 15, no. 1, pp. 6418-6426, 2016.

[3] T. T. K. M. M. Jahangir and X. S., "Diagnosis of cardiovascular diseases using artificial intelligence techniques: A review," International Journal of Computer Applications, vol. 183, no. 3, pp. 1-25, 2021.

[4] A. Syahid, S. Ali and S. Roselina, "Hybrid artificial neural network with artificial bee colony algorithm for crime classification," in Computational Intelligence in Information Systems, Springer, 2015, pp. 31-40.

[5] M. Sung-Hwan, L. Jumin and H. Ingoo, "Hybrid genetic algorithms and support vector machines for bankruptcy prediction," Expert systems with applications, vol. 31, no. 3, pp. 652-660, 2006.

[6] C. Tanujit,. C. Swarup and C. Ashis Kumar, "A novel hybridization of classification trees and artificial neural networks for selection of students in a business school," Opsearch, vol. 55, no. 2, pp. 434-446, 2018.

[7] C. Panayiotis,. C. Andreas and A. Andreas S, "A Hybrid Prediction Model Integrating Fuzzy Cognitive Maps with Support Vector Machines," in ICEIS (1), 2017, pp. 554-564.

[8] C. R. Abrunhosa,. P. Leonardo Antonio Monteiro, B. Laura, B. d. B. and R. Am'alia Faria dos, "A comparative study between artificial neural network and support vector machine for acute coronary syndrome prognosis," Pesquisa Operacional, vol. 36, no. 2, pp. 321-343, 2016.

[9] Y. B, W. YT, Y. JB and W. JY, "A comparison of the performance of ANN and SVM for the prediction of traffic accident duration," Neural Network World, vol. 26, no. 3, p. 271, 2016.

[10] S. Amit Kumar, P. Sudesh Kumar and. A. Mohammed, "A comparative study between naive Bayes and neural network (MLP) classifier for spam email detection," Int. J. Comput. Appl, 2014.

[11] Y. Bin, Y. Zhong-Zhen, C. Kang and Y. Bo, "Hybrid model for prediction of bus arrival times at next station," Journal of Advanced Transportation, vol. 44, no. 3, pp. 193-204, 2010.

[12] A.-K. Ahmad and. H. Haneen, "Classifying Diabetes Disease Using Feedforward MLP Neural Networks," in Technological Innovations in Knowledge Management and Decision Support, 2019, pp. 127-149.

[13] T. Divya and. A. Sonali, "A survey on Data Mining approaches for Healthcare," International Journal of Bio-Science and Bio-Technology, vol. 5, no. 5, pp. 241-266, 2013.

[14] V. Vladimir, The nature of statistical learning theory, Springer science & business media, 2013.

[15] R. Ms Jayshree S, M. Mr Prafulla L and P. Ms Dipali P, "A Review Paper on Classification of Stem Cell Transplant to Identify the High Survival Rate," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3, no. 4, pp. 1918-1920, 2015.

[16] O. Fernando EB, F. A. A and J. C. G, "A new sequential covering strategy for inducing classification rules with ant colony algorithms," IEEE Transactions on Evolutionary Computation, vol. 17, no. 1, pp. 64-76, 2012.

[17] V. V, "Statistical learning theory new york," NY: Wiley, 1988.

[18] A. S. Lua,. S. Zyad and K. Basel, "Data mining: A preprocessing engine," Journal of Computer Science, vol. 2, no. 9, pp. 735-739, 2006.