healthcare

MDPI

Article

A Fusion-Based Machine Learning Approach for the Prediction of the Onset of Diabetes

Muhammad Waqas Nadeem [1], Hock Guan Goh [1], Vasaki Ponnusamy [1], Ivan Andonovic [2,*], Muhammad Adnan Khan [3,*] and Muzammil Hussain [4]

[1] Faculty of Information and Communication Technology (FICT), Universiti Tunku Abdul Rahman (UTAR), Kampar 31900, Perak, Malaysia; waqasnadeem@utar.my (M.W.N.); gohhg@utar.edu.my (H.G.G.); vasaki@utar.edu.my (V.P.)
[2] Department of Electronic & Electrical Engineering, University of Strathclyde, Royal College Building, 204 George St., Glasgow G1 1XW, UK
[3] Pattern Recognition and Machine Learning Lab, Department of Software, Gachon University, Seongnam 13557, Korea
[4] Department of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore 54000, Pakistan; muzammil.hussain@umt.edu.pk
* Correspondence: i.andonovic@strath.ac.uk (I.A.); adnan@gachon.ac.kr (M.A.K.)

Abstract: A growing portfolio of research has been reported on the use of machine learning-based architectures and models in the domain of healthcare. The development of data-driven applications and services for the diagnosis and classification of key illness conditions is challenging owing to issues of low volume, low-quality contextual data for the training, and validation of algorithms, which, in turn, compromises the accuracy of the resultant models. Here, a fusion machine learning approach is presented reporting an improvement in the accuracy of the identification of diabetes and the prediction of the onset of critical events for patients with diabetes (PwD). Globally, the cost of treating diabetes, a prevalent chronic illness condition characterized by high levels of sugar in the bloodstream over long periods, is placing severe demands on health providers and the proposed solution has the potential to support an increase in the rates of survival of PwD through informing on the optimum treatment on an individual patient basis. At the core of the proposed architecture is a fusion of machine learning classifiers (Support Vector Machine and Artificial Neural Network). Results indicate a classification accuracy of 94.67%, exceeding the performance of reported machine learning models for diabetes by ~1.8% over the best reported to date.

Keywords: diabetes prediction; machine learning; support vector machines; artificial neural networks; data fusion; healthcare applications; intelligent system

Citation: Nadeem, M.W.; Goh, H.G.; Ponnusamy, V.; Andonovic, I.; Khan, M.A.; Hussain, M. A Fusion-Based Machine Learning Approach for the Prediction of the Onset of Diabetes. *Healthcare* **2021**, *9*, 1393. https://doi.org/10.3390/healthcare9101393

Academic Editor: Daniele Giansanti

Received: 6 September 2021
Accepted: 9 October 2021
Published: 18 October 2021

1. Introduction

Diabetes (DB) is a damaging health condition placing a significant treatment cost burden on health service providers throughout the world. Beta cells in the pancreas produce an insufficient amount of insulin with the resultant deficiency causing high levels of glucose within the blood, classified as Type-1 DB (hyper-glycemia); in Type-2, the body is unable to utilize the available insulin [1]. Moreover, DB gives rise to other clinical complications such as neurological damage, retinal degradation, and kidney and heart disease [2].

The treatment of DB is also an escalating challenge as more than 422 million adults suffered from the condition in 2014 compared to 108 million in 1980; the ratio of people-with-diabetes (PwD) referenced to the total adult population increased from 4.7% to 8.5% over the same period. Furthermore, 1.6 million diabetic patients died in 2015, and in 2012, 2.2 million further deaths were attributed to high blood glucose levels [3]. Projections indicate that DB will be the 7th major illness condition causing deaths in the global population by 2030 [4]. The timely identification and the early detection of the onset of diabetes are, therefore, of

potential value in the goal of optimizing treatment pathways, providing a better quality of life for PwD, and reducing the number of deaths owing to the condition. Moreover, a significant number of PwD remain unaware of the condition until a serious complication event [4]; delays in the diagnosis of Type-2 DB during the early stages of onset increases the risk of serious complications [1,4].

A range of Machine Learning (ML) methods such as Logistic Adaptive Network-based Fuzzy Inference System (LANFIS) [5], Q-learning Fuzzy ARTMAP (FAM), Genetic Algorithm (GA) (QFAM-GA) [6], Hybrid Prediction Model (HPM) [7], Artificial Neural Network (ANN), and Bayesian Networks (BN) (ANN-BN) [8] have been used to develop algorithms for the classification of DB [9,10]. However, reported machine learning-based solutions have been limited in the accuracy of prediction, owing primarily to the lack of the required scope and volume of data for the training and testing of models. Intuitively, algorithms generated from large datasets yield more accurate performance as compared to models trained with a small number of instances of the target output. Poorly performing machine learning algorithms increase the barrier to adoption in operational settings, and thus, there is a clear need to enhance the prediction accuracy given the limited availability and accessibility of the required scope of data [11–16].

Here, core elements of a decision support system founded on the fusion of machine learning algorithms are reported for the identification of diabetes. The proposed architecture comprises multiple layers segmented as Data Sources, Data Fusion, and Fusion of Machine Learning techniques [17]. The Data Source layer receives and stores multiple streams of data from different sources such as Electronic Health Records (EHR) in a format suitable as an input to the development of the machine learning algorithms. Subsequently, the Data Fusion layer fuses data from the Data Source layer, storing the outputs on a centralized database. K fold validation is then applied for the selection of hyper-parameters, inputs to the training of the classification model. In the development reported here, Support Vector Machine (SVM) and Artificial Neural Network (ANN) machine learning classifiers have been fused using the posteriori probability method for the classification of diabetes.

A range of performance metrics, viz. Accuracy, Specificity, Sensitivity, and Precision, are used in the evaluation of the performance of the proposed approach. Diabetic datasets reported in [18,19] are used for the training and testing of the models.

The contributions of the reported research can be summarized as:

- A fusion-based machine learning architecture for the prediction of diabetes has been proposed.
- Two machine learning classifiers Support Vector Machine (SVM) and Artificial Neural Network (ANN) within the architecture have been evaluated.

The remainder of the paper is organized as follows. Section 2 summarizes related research, Section 3 describes the methodology of the proposed architecture, the theoretical and mathematical background of the selected machine learning classifiers, and the approach to their fusion are also presented in this section. The details of the evaluation of the performance of the approach are mapped and discussed in Section 4, and conclusions are drawn in Section 5.

2. Related Research

Artificial Neural Networks (ANN) and General Regression Neural Networks (GRNN) have been used to create algorithms for the diagnosis of diabetes [20]; the GRNN approach achieved an ~80% prediction accuracy, an improvement compared to Radial Basis Function (RBF) and Multi-Layer Perceptron (MLP) based techniques. Temurtas et al. [21] reported on a Multilayer Neural Network (MNN) implementation deploying a probabilistic Neural Network and Levenberg–Marquardt Algorithm (LMA). A two-stage approach proposed in [22] achieved a ~89.5% prediction accuracy; in Stage-I, a Principal Component Analysis (PCA) algorithm is applied to reduce the dimension of input features and in Stage-II, an Adaptive Neuro-Fuzzy Inference System generates the model. Further, an Adaptive Network-based Fuzzy System (ANFS) and Levenberg–Marquardt Algorithm

(LMA)-based solution achieved an 82.3% prediction accuracy [23]. Rohollah et al. report on a Logistic Adaptive Network-based fuzzy system with 88% predictive accuracy [5] and Kemal et al. [24] developed a Least Square Support Vector Machine (LS-SVM) and Generalization Discriminant Analysis (GDA)-based cascade learning system. A k-means clustering approach reported by Bankat et al. [7] successfully eliminates incorrect samples from the dataset. Bayesian Network (BN) based diagnosis achieved a 72.3% prediction accuracy [25], whilst a three-stage diagnosis system presented by Muhammad et al. [26] uses a Genetic Algorithm (GA); several rule-based classification systems have been developed by the same research team. The rule-based system of Wiphada et al. [27] comprises two stages; in the first stage, the nodes of a neural network are pruned to determine their maximum weights; in the second stage, the data are analyzed to identify the frequency content, and then linguistic rules are created based on frequency intervals. The rule-based system has a 74% prediction accuracy. Mostafa et al. [28] present a Recursive Rule Extraction (Re-Rx) framework to generate decision rules, achieving 83.8% accuracy. In [6], a two-stage hybrid model was presented for decision rule extraction and classification. In stage-1, fuzzy logic with Q-learning is used to create decision rules and in stage-2, a Genetic Algorithm (GA) is used for the extraction of rules. Mohammad et al. [29] present a combination of Support Vectors Regression (SVR) and an ANN-based model for the detection of diabetes with 86.13% accuracy. A Gaussian Hidden Markov Model (GHMM) technique is applied in [30], achieving 85.69% accuracy; a Gaussian Hidden Markov Model (GHMM) reported in [31] achieved 85.9% accuracy; and a Deep Extreme Learning Machine (DELM) based prediction model is presented in [32] with 92.8% accuracy. A summary of the related research is presented in Table 1.

Table 1. Summary of the recent development of Machine Learning for Diabetic Prediction.

Studies	Proposed Methods	Dataset	Findings
[5]	Logistic Adaptive Network Fuzzy Inference System (LANFIS)	Pima Indians diabetes	Prediction accuracy = 88.05% Sensitivity = 92.15% Specificity = 81.63%
[7]	Hybrid Prediction Model (HPM)+ C 4.5	Pima Indian diabetes	Prediction accuracy = 92.38%
[20]	Artificial Neural Networks (ANN) + General Regression Neural Networks (GRNN)	Pima Indian diabetes	Prediction accuracy = 80%
[22]	Principal Component Analysis (PCA) + Adaptive Neuro-Fuzzy Inference System (ANFIS)	Pima Indian diabetes	Prediction accuracy = 89.47%
[23]	Adaptive Network-based Fuzzy System (ANFS) + Levenberg–Marquardt Algorithm	Pima Indian diabetes	Prediction accuracy = 82.30% Sensitivity = 66.23% Specificity = 89.78%
[24]	Least Square Support Vector Machine (LS-SVM) and Generalization Discriminant Analysis (GDA)	Pima Indian diabetes	Classification accuracy = 82.05% Sensitivity = 83.33% Specificity = 82.05%
[25]	Bayesian Network (BN)	Pima Indian diabetes	Prediction accuracy = 72.3%
[26]	(1) Genetic Algorithm (GA) + K-Nearest Neighbors (GA-KNN), (2) Genetic Algorithm (GA) + Support Vector Machine (GA-SVM)	Pima Indian diabetes	Prediction accuracy = 80.5%, Prediction accuracy = 87.0%,
[31]	Gaussian Hidden Markov Model (GHMM)	CPCSSN clinical dataset	Prediction accuracy = 85.9%
[32]	Deep Extreme Learning Machine (DELM)	Pima Indian diabetes	Prediction accuracy = 92.8%
[33]	Gradient Boosted Trees (GBTs)	Canadian AppleTree and the Israeli Maccabi Health Services (MHS)	Prediction accuracy = 92.5%
	Proposed SVM-ANN		Prediction accuracy = 94.67% Sensitivity = 89.23% Specificity = 97.32%

In summary, a significant body of research has been reported over the recent past detailing a range of machine learning approaches for the identification of diabetes and

prediction of the onset of critical episodes in PwD. Informed by the reported advances to date, the architecture detailed here implements a fusion-based approach to improve the prediction accuracy.

3. Materials and Methods

3.1. Datasets

Two datasets are used in the training and testing of the proposed fusion-based machine learning architecture. The first dataset is derived from the publicly available National Health and Nutrition Examination Survey (NHANES) [18], consisting of 9858 records and 8 features. The second "Pima Indian diabetes " [19] is acquired from the online repository "Kaggle", which comprises 769 records and 8 features. Both datasets, consisting of the same features but comprising a different number of records, are detailed in Table 2. Thus, the fused dataset has 10,627 records with 8 features with an age distribution between 21–77 years. The binary response attribute takes the values '1' or '0', where '0' means a non-diabetic patient and '1' means a diabetic patient. There are 7071 (66.53%) cases in class '0' and 3556 (33.46%) cases in class '1'.

Table 2. Diabetes Datasets—Features Description.

S#	Feature Name	Description	Variable Type
1	Glucose (F1)	Plasma glucose concentration at 2 h in an oral glucose tolerance test	Real
2	Pregnancies (F2)	Number of times pregnant	Integer
3	Blood Pressure (F3)	Diastolic blood pressure (mm HG)	Real
4	Skin Thickness (F4)	Triceps skinfold thickness (mm)	Real
5	Insulin (F5)	2-h serum insulin (mu U/mL)	Real
6	BMI (F6)	Body mass index (weight in kg/(height in)^2	Real
7	Diabetes Pedigree Function (F7)	Diabetes Pedigree Function	Real
8	Age (F8)	Age (years)	Integer

3.2. System Architecture

The architecture consists of the following layers designated as 'Data Source', 'Data Fusion', 'Pre-processing', 'Application', and 'Fusion'. The end-to-end process flow is described in Table 3, and the system architecture is depicted in Figure 1. The following is the methodology for the development of the algorithm.

Table 3. Steps for the Implementation of the Proposed Architecture.

1	Begin
2	Input Data
3	Apply Data fusion technique
4	Preprocess the data by different techniques
5	Data partitioning using the K-fold cross-validation method
6	Classification of diabetes and healthy peoples using SVM and ANN
7	Fusion of SVM and ANN
8	Computes performance of the architecture using a different evaluation matrix
9	Finish

3.2.1. Data Fusion

Data Fusion is a process of association and combination of data from multiple sources [15,34], characterized by continuous refinements of its estimates, evaluation of the need for additional data, and modification of its process to achieve improved data quality. Hall et al. [35] state that the fusion of data enables the development of methods for the semi-automatic or automatic transformation of multiple sources of information from different locations and times to support effective decision-making.

Several methods such as Luo and Key [36] and Dasarathy [37] have been reported in recent years and here, the latter has been selected as it has been proven to be the most efficient in fusing data [37]. Data In-Data Out (DAI-DAO), the most elementary function in the fusion process, accepts data from the input layer and cleans the data to be more aligned to the needs of the development of machine learning algorithms.

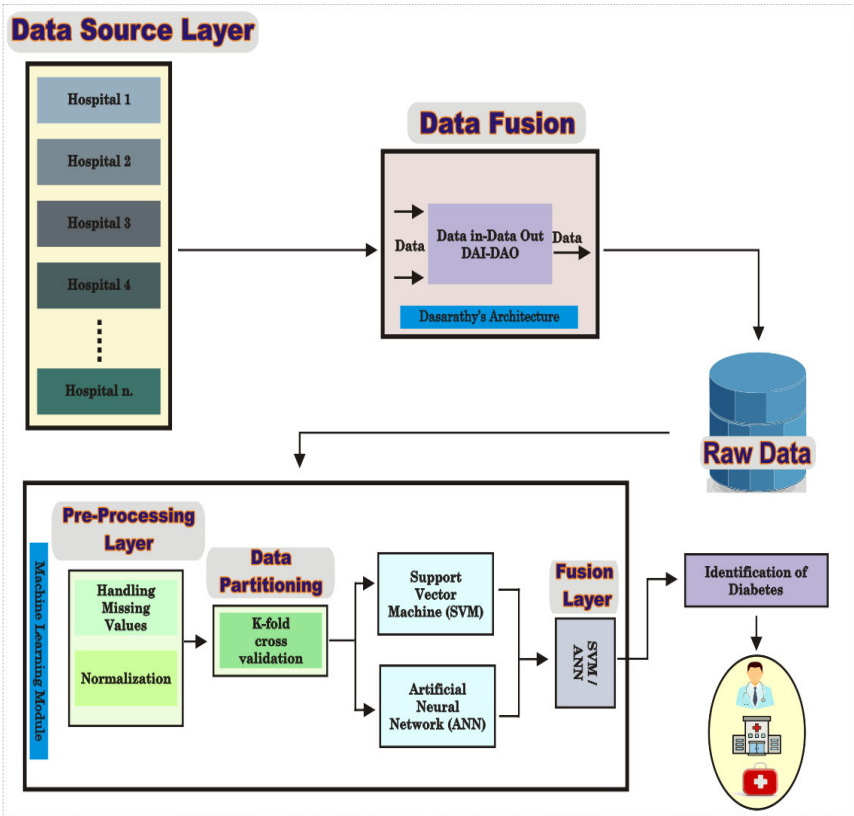

Figure 1. The System Architecture of the Proposed Fusion-based Prediction Approach.

Following the Dasarathy approach, Equation (1) represents the different data blocks $x_1, x_2, x_3 \ldots\ldots\ldots x_n$ and the output X:

$$Set\ X = \{x_1, x_2, x_3 \ldots\ldots\ldots x_n\} \tag{1}$$

The degree of support, A, is indicated by the proposition of Basic Probability Assignment (BPA); the greater the BPA, the greater the degree of support for A (Equation (2)). The combination of different BPAs is used to reach decisions on the optimum fusion of data:

$$m(X) = m_1 \otimes m_2 \otimes m_3 \ldots\ldots \otimes m_n = \left\{ \frac{1}{1-k} \sum A_1 \cap A_2 \cap A_3 \ldots\ldots \cap A_n = X^{m_1(A_1)m_2(A_2)\ldots m_n(A_n)} \right\} \tag{2}$$

The probability of conflict—referred to as the minimum distance between data points—is captured in Equation (3), where K represents the probability of conflict, which is computed as:

$$K = \sum_{A_1 \cap A_2 \cap A_3 \ldots\ldots \cap A_n = \varnothing} m_1(A_1)m_2(A_2)\ldots m_n(A_n) \tag{3}$$

3.2.2. Pre-Processing

Pre-processing is initiated by the treatment of missing values (P) followed by standardization (S). Missing or null values are imputed; otherwise, the accuracy of prediction

of the machine learning classifier is compromised [32]. Here, the mean method—instead of dropping—is used to fill the missing values, formulated as in Equation (4):

$$P(x) = \begin{cases} mean(x), & if\ x = null/missing \\ x, & otherwise \end{cases} \quad (4)$$

where x is the instances of the feature vector, which lies in n-dimensional space. The imputation of missing values by the mean method is warranted as it produces the required continuous data for the training of the algorithm without introducing outliers.

Standardization or Z-score normalization is used to rescale features, in so doing achieving a standard normal distribution with unit variance and zero mean. Standardization (S), formulated as in Equation (5), also reduces the skewness of the data distribution:

$$S(x) = \frac{x - x^{-}}{\propto} \quad (5)$$

where x is the n-dimensional instances of the feature vector, $x \in R^n$; $x^{-} \in R^n$ and $\propto \in R^n$ are the standard deviation and mean of attributes.

3.2.3. Cross-Fold Validation

The K-fold Cross-Validation (KCV) is a common approach used for model selection, error estimation of the classifiers, and splitting of data [38]. The dataset is partitioned into 5-folds; K-1 folds are used for the training and the fine-tuning of the hyper-parameters in the inner loop where the grid search algorithm was deployed [39]. In the outer loop (k times), the test data and optimum hyper-parameters were used for the evaluation of the model. The raw dataset contains the imbalanced ratio of negative and positive samples; a stratified KCV [40] has been used to maintain the same percentage of the samples for each class as in the original percentage. Equation (6) is used for the estimation of the final performance:

$$M = \frac{1}{K}\sum_{n=1}^{K} P_n \pm \sqrt{\frac{\sum_{n=1}^{K}(P_n - P^{-})^2}{K-1}} \quad (6)$$

where M is designated as the final performance metric for the classifier and $P_n \in R$ $n = 1, 2, 3, \ldots\ldots K$ is the performance metric for each fold.

3.2.4. Support Vector Machines

The SVM algorithm has been used extensively for classification, regression, identification, density estimation, and time series analysis. SVM models segment the data into different groups comprising data points with similar properties. Furthermore, the fundamental principle of SVMs is to compute the optimal hyper-planes that generate the best generalization of the dataset [41–43]. For example, in a linear SVM model, the inputs are separated from the non-linear mapping into a high-dimensional space. The linear model constructs a new space, which represents a nonlinear decision limit between the original and the new space.

The SVM model predicts the classes for a new sample. Given a training dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in R_n$ and y {+1,−1} where x_i represents the transferred input vector and y_i the target value, the SVM becomes a binary classifier in which the class labels feature only two values +1 or −1. SVM draws an optimal hyper-plane H that separates the data into different classes and the hyper-plane H from the inputs. The objective function has convexity, a significant advantage as the solution of a quadratic programming problem and the training of SVMs are equivalent, yielding a unique solution. In contrast, the Artificial Neural Network (ANN) method requires nonlinear optimization, which may result in the algorithm being held hostage to local minimums.

The precision of the SVM algorithm is greater than other reported forecasting methods. The SVM minimizes the structural risk, while other machine learning methods focus on

empirical risk minimization. In other words, the SVM method focuses on minimizing the upper limit of the generalization error to reduce the training error. SVMs process a large volume of data efficiently without overfitting. The SVM method also emphasizes the establishment of optimal hyperplanes for the separation of data. The training points $(x_i \rightarrow y_i)$ that are closet to the optimal hyperplanes are referred to as support vectors and also develop the limit of the decision planes. In general, in cases when the data are not separated linearly, the SVM method uses non-linear machines to trace the optimal hyperplanes that reduce the error rate in the training set of the data [44]. The core of the SVM method theory for the solving of binary classification is described in [42,43].

Consider a set of training points, $D = \{x_i, y_i\}_{i=1}^{N}$, where the input vectors are $x_i = (x^{(1)}, \ldots\ldots x^{(n)})\ E\ R^n$ and output vectors $y_i\ E\ \{0,1\}$, and where n represents the amount of training data. Then, the optimal hyperplane used to separate the classes of data points and these optimal hyperplanes are identified by solving the following optimization problem:

$$\underset{w,b}{Min}\left(\frac{1}{2}w^t w\right) \tag{7}$$

$$Subject: y_i\left(w^t \Phi(x_i) + b \geq 1\right),\ i = 1, 2, 3, \ldots\ldots, n$$

where w is the Wright vector and b represents a bias variable. The non-linear function $\Phi(.) : R^n \rightarrow R^{nk}$ maps the given inputs into a high dimensional space.

However, numerous classification problems are linearly non-separable; thus, $£_i$ denotes a gap variable used for misclassification. Hence, the optimization problem with the gap variable is written as:

$$\underset{w,b,£}{Min}\left(\frac{1}{2}w^t w + C\sum_{i=1}^{n} £_i\right) \tag{8}$$

$$Subject: \begin{cases} y_i\left(\left(w^t \Phi(x_i) + b\right)\right) + & £_i \geq 1,\ i = 1, 2, 3, \ldots\ldots, n \\ & £_i \geq 0,\ i = 1, 2, 3, \ldots\ldots, n \end{cases}$$

where C is used as a penalty variable for the error.

The Lagrangian construction function is used to solve the primary problem, and linear equality bound constraints are used to convert the primal into a quadratic optimization problem:

$$Max_a\left(\sum_{i=0}^{N} a_i - \frac{1}{2}\sum_{i=0}^{n}\sum_{j=0}^{n} a_i a_j Q_{\mathrm{ij}}\right)$$

$$Subject: \begin{cases} 0 \leq a_i < C,\ i = 1, 2, 3, \ldots\ldots, n \\ \sum\limits_{i=0}^{N} a_i y_i = 0 \end{cases}$$

where a_i is known as Lagrange multiplier $Q_{ij} = y_i y_j\ \Phi(x_i)^t \Phi(x_j)$.

The kernel function not only replaces the internal product but also satisfies the Mercer condition $K_{(x_i, x_j)} = \Phi(x_i)^t \Phi(x_j)$, used for the representation of proximity or similarity between data points. Finally, the non-linear decision function is used in the primal space for the linearly non-separable case:

$$y(x) = sgn\left(\sum_{i=0}^{N} a_i y_i K(x_i,\ x_j) + b\right)$$

The kernel function maps input data into a large dimensional space, where hyperplanes separate the data, rendering the data linearly separable. Different kernel functions are potential candidates for use by the SVM method:

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

(ii) Radical Kernel: $K(x_i, x_j) = exp(-\gamma \mid |x_i - x_j| \mid^2)$

(iii) Polynomial Kernel: $K(x_i, x_j) = (yx_i^T x_j + r)^d$

(iv) Sigmoid Kernel: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$, where $r, d \in N$ and $\gamma \in R^+$ all are constants.

The kernel functions play an important role when the complex decision limits are defined between different classes. The selection of the decision limits is critical and challenging; hence, the selection of potential mappings is the first task for a given classification problem. The optimal selection of the potential mapping minimizes generalization errors.

In the reported research, the Radial Basis Function (RBF) kernel is selected most often for the creation of a high dimensional space for the non-linear mapping of samples. Furthermore, the RBF kernel treats non-linear problems more easily as compared to the Linear kernel. The Sigmoid kernel is not valid for some parameters.

The second challenge is the selection of hyperparameters that impact the complexity of the model. The Polynomial kernel has more hyperparameters as compared to the RBF kernel, but the latter is less computationally intensive during the Polynomial kernel, requiring more computational time at the training phase.

3.2.5. Artificial Neural Networks

Artificial Neural Networks (ANNs) are inspired by the structure and functional aspects of the human biological neural system. The ANN method originates from the field of computer science, but the applications of ANNs are now widely used within a growing number of research disciplines [45]; the combination of large amounts of unstructured data ('big data') coupled to the versatility of the ANN architecture have been harnessed to obtain ground-breaking results in numerous application domains including natural language processing, speech recognition, and detection of autism genes. ANNs comprises many groups of interconnected artificial neurons executing computations through a connectionist approach.

Typical ANN architectures are composed of three types of nodes, viz. input, hidden, and output. The former contains the explanatory parameters and the level of attributes varies from model to model. The dependent variables are contained by the output nodes and the number of output nodes depends on choice probabilities. Nodes are connected through links and the signals propagate in a forward direction. Different numerical weights are computed from the data assigned to each link. At each node, the input value of the previous node is multiplied by the weight and summed. An activation function is used to propagate the signal into the next layer; activation functions 'SoftMax', 'tan-sigmoid', and 'purlin' have been used commonly in ANNs architectures. The sigmoid activation function is used here. Weigh initialization, feedforward, backpropagation for error, updating weights, and bias are integral to the ANNs.

The algebraic formulation of ANNs is:

$$f_j = b_1 + \sum_{i=1}^{n} (w_{ij} * r_i) \tag{9}$$

where the w_{ij} represents the weight of neurons, r_i represents the inputs, and b is the bias. Further, the 'sigmoid' activation function is written as:

$$\varphi_k = \frac{1}{1 + e^{-f_j}} \text{ where } k = 1, 2, 3 \ldots r \tag{10}$$

Equation (10) is used to compute the error in back-propagation:

$$E = \frac{1}{2} \sum_{k} (\tau_k - \varphi_k)^2$$

where the τ_k denotes the desired output and φ_k represents the calculated output. Thus, the rate of change in weights is calculated as:

$$\Delta w \propto -\frac{\partial E}{\partial w}$$

$$\Delta v_{j,k} = -\epsilon \frac{\partial E}{\partial v_{j,k}}$$

Equation (11) describes the updating of weights and biases between the hidden and output layers. By using the chain rule:

$$\Delta v_{j,k} = -\epsilon \frac{\partial E}{\partial \varphi_k} \times \frac{\partial \varphi_k}{\partial \psi_k} \times \frac{\partial \psi_k}{\partial v_{j,k}}$$

$$\Delta v_{j,k} = \epsilon(\tau_k - \varphi_k) \times \varphi_k(1 - \varphi_k) \times \left(\varphi_j\right)$$

$$\Delta v_{j,k} = \epsilon \xi_k \varphi_j$$

$$\xi_k = (\tau_k - \varphi_k) \times \varphi_k(1 - \varphi_k)$$

$$\Delta w_{i,j} \propto -\left[\sum_k \frac{\partial E}{\partial \varphi_k} \times \frac{\partial \varphi_k}{\partial \psi_k} \times \frac{\partial \psi_k}{\partial \varphi_j}\right] \times \frac{\partial \varphi_j}{\partial \psi_j} \times \frac{\partial \psi_j}{\partial w_{i,j}}$$

$$\Delta w_{i,j} = -\epsilon \left[\sum_k \frac{\partial E}{\partial \varphi_k} \times \frac{\partial \varphi_k}{\partial \psi_k} \times \frac{\partial \psi_k}{\partial \varphi_j}\right] \times \frac{\partial \varphi_j}{\partial \psi_j} \times \frac{\partial \psi_j}{\partial w_{i,j}}$$

$$\Delta w_{i,j} = \epsilon \left[\sum_k (\tau_k - \varphi_k) \times \varphi_k(1 - \varphi_k) \times \left(v_{j,k}\right)\right] \times \varphi_k(1 - \varphi_k) \times r_i$$

$$\Delta w_{i,j} = \epsilon \left[\sum_k (\tau_k - \varphi_k) \times \varphi_k(1 - \varphi_k) \times \left(v_{j,k}\right)\right] \times \varphi_j\left(1 - \varphi_j\right) \times r_i$$

$$\Delta w_{i,j} = \epsilon \left[\sum_k \xi_k\left(v_{j,k}\right)\right] \times \varphi_j\left(1 - \varphi_j\right) \times r$$

$$\Delta w_{i,j} = \epsilon \xi_j \, r_i$$

where

$$\xi_j = \left[\sum_k \xi_k\left(v_{j,k}\right)\right] \times \varphi_j\left(1 - \varphi_j\right) \tag{11}$$

Similarly, Equation (12) describes the updating of weight and bias between hidden and input layers:

$$v_{j,k}^{+} = v_{j,k} + \lambda_F \Delta v_{j,k}$$

$$w_{i,j}^{+} = w_{i,j} + \lambda_F \Delta w_{i,j} \tag{12}$$

where λ_F represents the learning rate.

3.2.6. Fusion of SVM-ANN

Traditional machine learning classifiers can be fused by different methods and rules [14]; the most commonly used fusion rules are 'min', 'mean', 'max', and 'product' [13]. $P_i(\varphi_j|x)$ represents the posteriori probability, most often applied to view the output of the classifiers, and it can also be used for the implementation of fusion rules. P_i represents the output of the i^{th}-classifier, φ_i represent the i^{th}-class of objects, and $P_i(x|\varphi_j)$ represents the probability

of x in the j^{th}-classifier given that the j^{th}-class of objects occured. As the proposed objective of the architecture is a two-class output, the posteriori probability can be written as:

$$P_i(\varphi_j|x) = \frac{P_i(x|\varphi_j)P(\varphi_j)}{P_i(x)}$$

$$P_i(\varphi_j|x) = \frac{P_i(x|\varphi_j)P(\varphi_j)}{P_i(x|\varphi_1)P(\varphi_1) + P_i(x|\varphi_2)P(\varphi_2)}$$
$$j = 1, 2 \text{ and } i = 1, 2, 3 \ldots\ldots, L$$

where L represents the number of classifiers; here, 2 classifers are selected, SVM & ANN. Thus, the posteriori probability for the target class can be written as:

$$P_i(\varphi_t|x) = \frac{P_i(x|\varphi_t)P(\varphi_t)}{P_i(x|\varphi_t)P(\varphi_t) + \theta_i \cdot P(\varphi_o)} \tag{13}$$

where φ_t represents the target class, φ_o is the outlier class, and θ_i is the uniform distribution of density for the feature set, and where $P(\varphi_t)$, $P(\varphi_o)$, and $P_i(x|\varphi_t)$ represent the probability of the target class, probability of the outlier class/miss predicted class, and probability of event x in the i^{th}-classifier given that the target class of object has occurred, respectively.

In the architecture, the mean fusion rule is applied, formulated as:

$$\mu(\varphi_t|x) = \frac{1}{L}\sum_{i=0}^{L} P_i(\varphi_t|x)$$

$$\mu(\varphi_o|x) = \frac{1}{L}\sum_{i=0}^{L} P_i(\varphi_o|x)$$

where $P_i(\varphi_t|x)$ and $P_i(\varphi_o|x)$ represent the probabilty of the target class given that x event has occurred in the i^{th}-classifier and the probabilty of the outlier class given that x event has occurred in the i^{th}-classifier. Then, the decision criteria are computed as:

$$\mu(\varphi_t|x) < \mu(\varphi_o|x) \text{ where } x \text{ is an outlier}$$

For j^{th}-class, the fusion rule can be written as:

$$\mu(\varphi_t|x) = \frac{1}{L}\sum_{i=0}^{L} P_i(\varphi_t|x)$$

After substitues, the values of $P_i(\varphi_t|x)$ are

$$\mu(\varphi_t|x) = \frac{1}{L}\sum_{i=0}^{L} \frac{P_i(x|\varphi_t)P(\varphi_t)}{P_i(x)}$$

If $P_i(x) \cong P(x)\forall_i$ then the above can be written as:

$$\mu(\varphi_t|x) = \frac{1}{L}\sum_{i=0}^{L} \frac{P_i(x|\varphi_t)P(\varphi_t)}{P(x)}$$

$$\mu(\varphi_t|x) = \frac{1}{L}\frac{P(\varphi_t)}{P(x)}\sum_{i=0}^{L} P_i(x|\varphi_t)$$

where,

$$Y_{\text{avg}}(x) = \frac{1}{L}\sum_{i=0}^{L} P_i(x|\varphi_t)$$

The target output is then computed as follows:

$$\theta^{-} = \frac{P(\varphi_o)}{P(\varphi_t)} \cdot \frac{1}{L} \sum_{i=0}^{L} \theta_i \tag{14}$$

The decision criteria, therefore, simplify as $Y_{avg}(x) < \theta^{-}$, where x is an outlier. Equation (14) represents the density function used to combine the class-conditional probability instead of posterior probabilities, which are estimated by each classifier. $Y_{avg}(x)$ is the final output of the architecture, and θ^{-} is used as a threshold that can be independently tuned to attain the desired trade-off between the false-negative rate and the false-positive rate.

4. Performance Evaluation

4.1. Performance Evaluation Matrix

A matrix comprising the accuracy, specificity, sensitivity, miss rate, precision, false-positive ratio, and the false-negative ratio is used to evaluate the performance of the algorithm [46]. A binary confusion matrix is used to compute the matrix. The development and evaluation of the solution has been performed in the Python 3.7 environment using a range of machine learning libraries on Intel® Core™ i3-3217U CPU @ 1.80 GHz PC.

The definition of the performance parameters within the matrix are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$$

$$\text{Miss rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$$

$$\text{Sensitivity} = \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%$$

$$\text{False positive ratio} = 1 - \left(\frac{\text{specificity}}{100}\right)$$

$$\text{False negative ratio} = 1 - \left(\frac{\text{sensitivity}}{100}\right)$$

4.2. Performance Results and Discussion

The performance of both classifiers has been evaluated as standalone and after fusion. A comparative analysis shows that the fusion of SVM and ANN provides an enhancement of the accuracy of prediction compared to the performance of SVM and ANN standalone algorithms. Results indicate a classification accuracy of 94.6%, exceeding the performance of machine learning models reported to date such as Random Forest (RF) [7] and Naïve Bayes (NB).

Figure 2a–c show the confusion matrix of ANN, SVM, and SVM-ANN, respectively. Figure 3 shows a class level comparison of the different machine learning methods, viz. SVM, ANN, and Fusion of SVM and ANN (SVM-ANN), which are used within the architecture. Results indicate that the SVM method yields a 93.02% accuracy for the negative class (healthy) and 78.62% accuracy for the positive class (diabetic); the ANN method: 97.21% for the negative and 86.29% for the positive class; and SVM-ANN: 97.32% for the negative and 89.23% for the positive class.

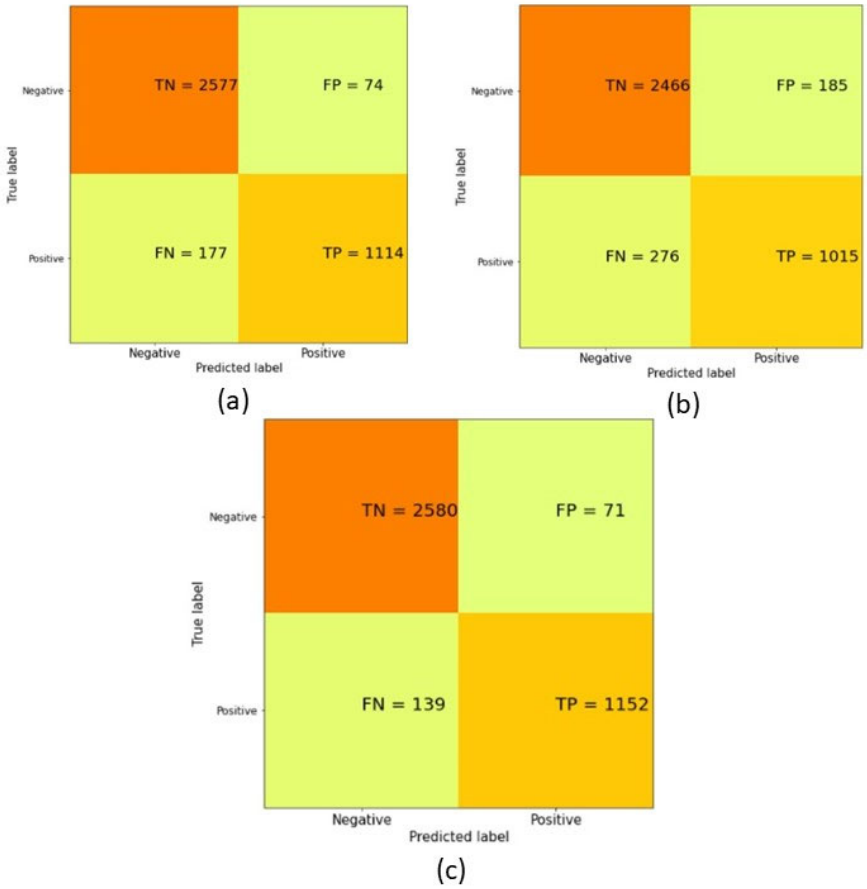

Figure 2. (**a**) Confusion matrix of ANN; (**b**) Confusion matrix of SVM; (**c**) Confusion matrix of Fusion (SVM-ANN).

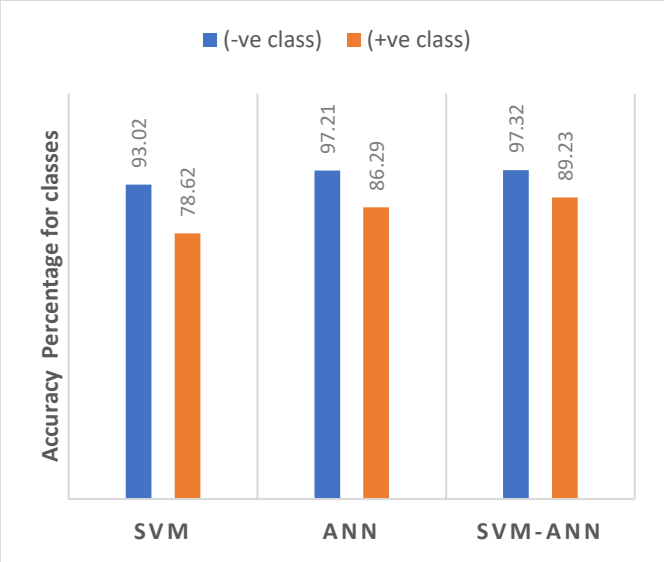

Figure 3. Class level Comparison of Accuracy.

The results of the comparative analysis of the performance of the architecture is also presented in Table 4. The SVM, ANN, and SVM-ANN approaches achieve 88.30%, 93.63%, and 94.67% accuracy, respectively, indicating an improvement in performance owing to the fusion of both.

Table 4. Overview of Simulation Results.

Evaluation Matrix	SVM	ANN	Fusion of SVM-ANN
Accuracy	88.30%	93.63%	94.67%
Specificity	93.02%	97.20%	97.32%
Sensitivity	78.62%	86.28%	89.23%
Precision	84.58%	93.77%	94.19%
Miss rate	11.70%	6.37%	5.33%
False Positive Ratio (FPR)	0.06	0.02	0.02
False Negative Ratio (FNR)	0.21	0.13	0.10

Figure 4 shows the comparison in terms of accuracy, precision, sensitivity, and specificity. On inspection, the SVM-ANN produces the best prediction in terms of accuracy in comparison with standalone SVM and ANN, improving the accuracy by a margin of 6.37% and 1.04%, respectively. The SVM-ANN model also improves the true-positive rate and also evident is that the degree of miss-classification is decreased by the SVM-ANN model. Furthermore, the fused architecture improves the system performance in terms of balanced accuracy, precision, sensitivity, and specificity (average of accuracy, precision, sensitivity, and specificity) by 3.71%, 4.98%, 6.78%, and 2.21%, respectively.

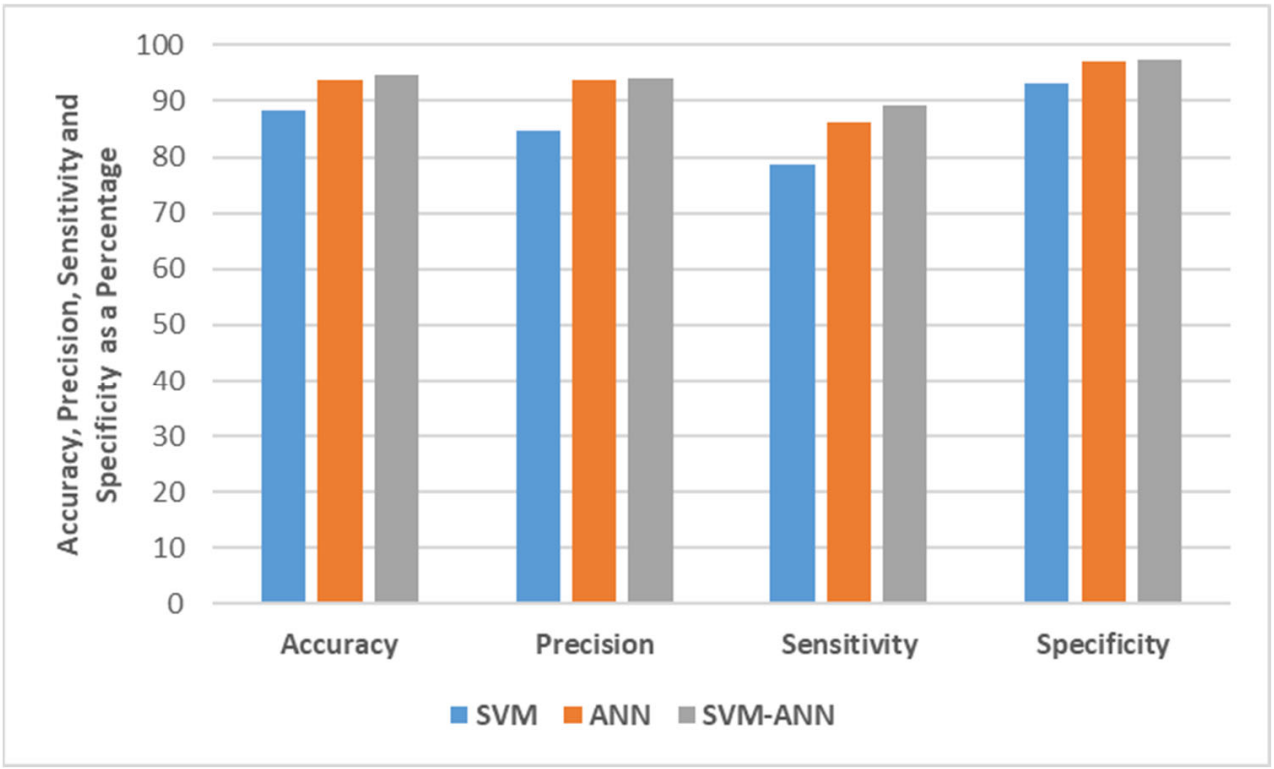

Figure 4. Comparison in terms of Accuracy, Precision, Sensitivity, and Specificity.

A comparison of the performance of the fusion-based system with models reported in the literature in terms of accuracy is shown in Figure 5. Results demonstrate that the architecture reported here yields an increase in the accuracy of the classification of diabetes with ~1.8% improvement compared to the best performing algorithm.

Figure 5. Performance Comparison of the Proposed Approach with existing Models in terms of Accuracy.

5. Conclusions

Machine learning methods and techniques are beginning to play an ever-increasing role in the domain of healthcare for the analysis of medical data to support the diagnosis and inform on the optimum treatment of critical health conditions. However, the existence and availability of sufficiently large datasets for the training and testing of machine learning models remain a barrier to achieving better-performing algorithms. Thus, given the prevailing restrictions in the scope and quality of available data, the paper reports on the development and evaluation of the performance of an approach based on the fusion of two machine learning methods for the prediction of diabetes.

The architecture performs data fusion to prepare a coherent dataset derived from multiple streams (locations) in order to be better aligned to the needs of the development of machine learning algorithms. The algorithms are then created through the fusion of two well-known machine learning classifiers: SVM and ANN, for the identification and prediction of the onset of critical events for PwD.

A comparison of performance with the published literature shows that the proposed architecture accuracy of 94.67% is an improvement of ~1.8% when compared to the best-performing model reported to date. In the future, other machine learning classifiers such as Random Forest, Decision Tree, and Naïve Base can also be consideredat the machine learning fusion layer.

Author Contributions: Conceptualization, M.W.N. and M.A.K.; methodology, M.W.N.; software, M.W.N.; validation, M.A.K., M.H. and H.G.G.; formal analysis, V.P.; investigation, I.A.; resources, M.H.; writing—original draft preparation, M.W.N. and M.H.; writing—review and editing, H.G.G. and I.A.; visualization, I.A.; supervision, H.G.G.and V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alberti, K.G.M.M.; Zimmet, P.; Shaw, J. International Diabetes Federation: A consensus on Type 2 diabetes prevention. *Diabet. Med.* **2007**, *24*, 451–463. [CrossRef]
2. Mellitus, D. Diagnosis and classification of diabetes mellitus. *Diabetes Care* **2005**, *28*, S5–S10.
3. WHO. *World Health Statistics 2016: Monitoring Health for the SDGs Sustainable Development Goals*; World Health Organization: Geneva, Switzerland, 2016.
4. Franciosi, M.; De Berardis, G.; Rossi, M.C.; Sacco, M.; Belfiglio, M.; Pellegrini, F.; Tognoni, G.; Valentini, M.; Nicolucci, A.; For the IGLOO Study Group. Use of the Diabetes Risk Score for Opportunistic Screening of Undiagnosed Diabetes and Impaired Glucose Tolerance: The IGLOO (Impaired Glucose Tolerance and Long-Term Outcomes Observational) study. *Diabetes Care* **2005**, *28*, 1187–1194. [CrossRef] [PubMed]
5. Ramezani, R.; Maadi, M.; Khatami, S.M. A novel hybrid intelligent system with missing value imputation for diabetes diagnosis. *Alex. Eng. J.* **2018**, *57*, 1883–1891. [CrossRef]
6. Pourpanah, F.; Lim, C.P.; Saleh, J.M. A hybrid model of fuzzy ARTMAP and genetic algorithm for data classification and rule extraction. *Expert Syst. Appl.* **2016**, *49*, 74–85. [CrossRef]
7. Patil, B.; Joshi, R.; Toshniwal, D. Hybrid prediction model for Type-2 diabetic patients. *Expert Syst. Appl.* **2010**, *37*, 8102–8108. [CrossRef]
8. Alic, B.; Gurbeta, L.; Badnjevic, A. Machine learning techniques for classification of diabetes and cardiovascular diseases. In Proceedings of the 2017 6th Mediterranean Conference on Embedded Computing (MECO), Bar, Montenegro, 11–15 June 2017; pp. 1–4. [CrossRef]
9. Nadeem, M.W.; Ghamdi, M.A.A.; Hussain, M.; Khan, M.A.; Khan, K.M.; Almotiri, S.H.; Butt, S.A. Brain tumor analysis empowered with deep learning: A review, taxonomy, and future challenges. *Brain Sci.* **2020**, *10*, 118. [CrossRef] [PubMed]
10. Nadeem, M.W.; Goh, H.G.; Ali, A.; Hussain, M.; Khan, M.A. Bone Age Assessment Empowered with Deep Learning: A Survey, Open Research Challenges and Future Directions. *Diagnostics* **2020**, *10*, 781. [CrossRef]
11. Fernández-Caramés, T.M.; Froiz-Míguez, I.; Blanco-Novoa, O.; Fraga-Lamas, P. Enabling the Internet of Mobile Crowdsourcing Health Things: A Mobile Fog Computing, Blockchain and IoT Based Continuous Glucose Monitoring System for Diabetes Mellitus Research and Care. *Sensors* **2019**, *19*, 3319. [CrossRef]
12. Dhillon, V.; Metcalf, D.; Hooper, M. Blockchain in healthcare. In *Blockchain-Enabled Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 201–220.
13. Yaqoob, I.; Salah, K.; Jayaraman, R.; Al-Hammadi, Y. Blockchain for healthcare data management: Opportunities, challenges, and future recommendations. *Neural Comput. Appl.* **2021**, *33*, 1–16.
14. Cichosz, S.L.; Stausholm, M.; Kronborg, T.; Vestergaard, P.; Hejlesen, O. How to Use Blockchain for Diabetes Health Care Data and Access Management: An Operational Concept. *J. Diabetes Sci. Technol.* **2018**, *13*, 248–253. [CrossRef]
15. Bhardwaj, R.; Datta, D. *Development of a Recommender System HealthMudra Using Blockchain for Prevention of Diabetes*; Scrivener Publishing LLC: Hoboken, NJ, USA, 2020; pp. 313–327. [CrossRef]
16. Bhavin, M.; Tanwar, S.; Sharma, N.; Tyagi, S.; Kumar, N. Blockchain and quantum blind signature-based hybrid scheme for healthcare 5.0 applications. *J. Inf. Secur. Appl.* **2021**, *56*, 102673.
17. Nadeem, M.W.; Goh, H.G.; Khan, M.A.; Hussain, M.; Mushtaq, M.F.; Ponnusamy, V.A. Fusion-Based Machine Learning Architecture for Heart Disease Prediction. *Comput. Mater. Contin.* **2021**, *67*, 2481–2496. [CrossRef]
18. Maniruzzaman, M.; Rahman, J.; Ahammed, B.; Abedin, M. Classification and prediction of diabetes disease using machine learning paradigm. *Health Inf. Sci. Syst.* **2020**, *8*, 7. [CrossRef]
19. Malik, S.; Harous, S.; El-Sayed, H. Comparative Analysis of Machine Learning Algorithms for Early Prediction of Diabetes Mellitus in Women. In Proceedings of the 2020, International Symposium on Modelling and Implementation of Complex Systems, Batna, Algeria, 24–26 October 2020; Springer: Cham, Switzerland, 2020; Volume 156, pp. 95–106. [CrossRef]
20. Kayaer, K.; Yildirim, T. Medical diagnosis on Pima Indian diabetes using general regression neural networks. In Proceedings of the International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP), Istambul, Turkey, 27 June 2003; Volume 181, p. 184.
21. Temurtas, H.; Yumusak, N.; Temurtas, F. A comparative study on diabetes disease diagnosis using neural networks. *Expert Syst. Appl.* **2009**, *36*, 8610–8615. [CrossRef]
22. Polat, K.; Güneş, S. An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. *Digit. Signal. Process.* **2007**, *17*, 702–710. [CrossRef]
23. Sagir, A.M.; Sathasivam, S. Design of a modified adaptive neuro fuzzy inference system classifier for medical diag-nosis of Pima Indians Diabetes. *AIP Conf. Proc.* **2017**, *1870*, 40048.
24. Polat, K.; Güneş, S.; Arslan, A. A cascade learning system for classification of diabetes disease: Generalized discri-minant analysis and least square support vector machine. *Expert Syst. Appl.* **2008**, *34*, 482–487. [CrossRef]
25. Guo, Y.; Bai, G.; Hu, Y. Using bayes network for prediction of type-2 diabetes. In Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions, London, UK, 10–19 December 2012; pp. 471–472.

26. Aslam, M.W.; Zhu, Z.; Nandi, A.K. Feature generation using genetic programming with comparative partner se-lection for diabetes classification. *Expert Syst. Appl.* **2013**, *40*, 5402–5412. [CrossRef]
27. Wettayaprasit, W.; Sangket, U. Linguistic knowledge extraction from neural networks using maximum weight and frequency data representation. In Proceedings of the Conference on Cybernetics and Intelligent Systems, Bangkok, Thailand, 7 June 2006; pp. 1–6. [CrossRef]
28. Ganji, M.F.; Abadeh, M.S. Using fuzzy ant colony optimization for diagnosis of diabetes disease. In Proceedings of the 18th Iranian Conference on Electrical Engineering, Isfahan, Iran, 11–13 May 2010; pp. 501–505. [CrossRef]
29. Beloufa, F.; Chikh, M. Design of fuzzy classifier for diabetes disease using Modified Artificial Bee Colony algorithm. *Comput. Methods Programs Biomed.* **2013**, *112*, 92–103. [CrossRef] [PubMed]
30. Zangooei, M.H.; Habibi, J.; Alizadehsani, R. Disease Diagnosis with a hybrid method SVR using NSGA-II. *Neurocomputing* **2014**, *136*, 14–29. [CrossRef]
31. Perveen, S.; Shahbaz, M.; Saba, T.; Keshavjee, K.; Rehman, A.; Guergachi, A. Handling Irregularly Sampled Longitudinal Data and Prognostic Modeling of Diabetes Using Machine Learning Technique. *IEEE Access* **2020**, *8*, 21875–21885. [CrossRef]
32. Rehman, A.; Athar, A.; Khan, M.A.; Abbas, S.; Fatima, A.; Saeed, A. Modelling, simulation, and optimization of dia-betes type II prediction using deep extreme learning machine. *J. Ambient Intell. Smart Environ.* **2020**, Preprint. 1–14.
33. Cahn, A.; Shoshan, A.; Sagiv, T.; Yesharim, R.; Goshen, R.; Shalev, V.; Raz, I. Prediction of progression from pre-diabetes to diabetes: Development and validation of a machine learning model. *Diabetes Metab. Res. Rev.* **2019**, *36*, e3252. [CrossRef]
34. White, F.E. *Data Fusion Lexicon*; Joint Directors of Labs: Washington, DC, USA, 1991.
35. Llinas, J.; Hall, D. An introduction to multi-sensor data fusion. In Proceedings of the IEEE International Symposium on Circuits and Systems (Cat. No. 98CH36187), Baltimore, MD, USA, 7–9 June 1998. [CrossRef]
36. Luo, R.C.; Kay, M. Multisensor Integration And Fusion: Issues And Approaches. *Proc. SPIE* **1988**, *0931*. [CrossRef]
37. Luo, R.C.; Yih, C.-C.; Su, K.L. Multisensor fusion and integration: Approaches, applications, and future research directions. *IEEE Sens. J.* **2002**, *2*, 107–119. [CrossRef]
38. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [CrossRef]
39. Krstajic, D.; Buturovic, L.J.; Leahy, D.E.; Thomas, S. Cross-validation pitfalls when selecting and assessing regression and classification models. *J. Chemin.* **2014**, *6*, 1–15. [CrossRef]
40. Zeng, X.; Martinez, T.R. Distribution-balanced stratified cross-validation for accuracy estimation. *J. Exp. Theor. Artif. Intell.* **2000**, *12*, 1–12. [CrossRef]
41. Awad, M.; Khanna, R. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*; Springer Nature: Berlin/Heidelberg, Germany, 2015.
42. Sands, T.M.; Tayal, D.; Morris, M.E.; Monteiro, S.T. Robust stock value prediction using support vector machines with particle swarm optimization. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC). IEEE, Cancun, Mexico, 20–23 June 2015; pp. 3327–3331. [CrossRef]
43. Sheta, A.F.; Ahmed, S.E.M.; Faris, H. A comparison between regression, artificial neural networks and support vector machines for predicting stock market index. *Soft Comput.* **2015**, *7*, 2.
44. Ding, Y.; Song, X.; Zen, Y. Forecasting financial condition of Chinese listed companies based on support vector ma-chine. *Expert Syst. Appl.* **2008**, *34*, 3081–3089. [CrossRef]
45. Luo, L.; Chen, X. Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction. *Appl. Soft Comput.* **2013**, *13*, 806–816. [CrossRef]
46. Van Cranenburgh, S.; Alwosheel, A. An artificial neural network based approach to investigate travellers' decision rules. *Transp. Res. Part. C Emerg. Technol.* **2019**, *98*, 152–166. [CrossRef]