electronics

MDPI

Article

A Comprehensive Medical Decision–Support Framework Based on a Heterogeneous Ensemble Classifier for Diabetes Prediction

Shaker El-Sappagh [1,2], Mohammed Elmogy [3], Farman Ali [1], Tamer ABUHMED [4], S. M. Riazul Islam [5] and Kyung-Sup Kwak [1,*]

1 Department of Information and Communication Engineering, Inha University, Incheon 22212, Korea; shaker_elsapagh@yahoo.com (S.E.-S.); farmankanju@gmail.com (F.A.)
2 Information Systems Department, Faculty of Computers and Informatics, Benha University, Banha 13518, Egypt
3 Information Technology Department, Faculty of Computers and Information, Mansura University, Mansura 35516, Egypt; elmogy@gmail.com
4 Computer Engineering Department, INHA University, Incheon 22212, Korea; tamer@inha.ac.kr
5 Department of Computer Science and Engineering, Sejong University, Seoul 05006, Korea; riaz@sejong.ac.kr
* Correspondence: kskwak@inha.ac.kr

Received: 19 May 2019; Accepted: 3 June 2019; Published: 5 June 2019

Abstract: Early diagnosis of diabetes mellitus (DM) is critical to prevent its serious complications. An ensemble of classifiers is an effective way to enhance classification performance, which can be used to diagnose complex diseases, such as DM. This paper proposes an ensemble framework to diagnose DM by optimally employing multiple classifiers based on bagging and random subspace techniques. The proposed framework combines seven of the most suitable and heterogeneous data mining techniques, each with a separate set of suitable features. These techniques are k-nearest neighbors, naïve Bayes, decision tree, support vector machine, fuzzy decision tree, artificial neural network, and logistic regression. The framework is designed accurately by selecting, for every sub-dataset, the most suitable feature set and the most accurate classifier. It was evaluated using a real dataset collected from electronic health records of Mansura University Hospitals (Mansura, Egypt). The resulting framework achieved 90% of accuracy, 90.2% of recall = 90.2%, and 94.9% of precision. We evaluated and compared the proposed framework with many other classification algorithms. An analysis of the results indicated that the proposed ensemble framework significantly outperforms all other classifiers. It is a successful step towards constructing a personalized decision support system, which could help physicians in daily clinical practice.

Keywords: diabetes mellitus; ensemble classifier; medical diagnosis; clinical decision support system

1. Introduction

Diabetes mellitus (DM) is a complex chronic disease [1]. It is estimated that in 2030 the incidence of diabetes will be 39% higher than it was in 2000 [2]. In 2013, around 382 million adults worldwide had DM, and it is predicted that there will be 592 million people with diabetes by 2035 [3]. DM is a primary source of morbidity and mortality. It contributes to increasing the risk of heart disease by two to four times [4]. The early detection and diagnosis of DM can help prevent and treat many complex complications and comorbidities. DM has an asymptomatic nature, especially in the early stages. As a result, a patient can have diabetes for 9 to 12 years before being diagnosed [5]. In most cases, the patient is already also affected by other complications at the time of diagnosis.

The massive volume of patient data collected from electronic health records (EHRs) makes an analysis of such data by hand inadequate and inaccurate, even if done by experts [6]. Experts have manually design algorithms based on their experience. These algorithms are increasingly proved limited and to not have scalability capabilities [4,7]. In addition, experts depend on a conservative identification strategy in algorithm design. Thus, they may fail to identify complex (e.g., borderline) patients, and can miss potential cases. Accuracy carries important weight in the medical domain because it concerns the lives of individuals [8]. Data mining prediction and classification techniques can be used to automate the discovery of hidden and potentially useful patterns in the massive volume of data [4]. Data mining can be defined as the process of discovering unknown patterns or relationships by selecting, exploring, and modeling large amounts of data [9]. Its basis includes statistics, machine learning, pattern recognition, database, and optimization techniques. It has a standard model, named the cross-industry process for data mining (CRISP-DM).

Recently, many classification algorithms based on EHR data have been used to enhance the detection of complex diseases, such as diabetes [7–12]. However, a few studies have used data mining techniques to build prediction models for diabetes diagnosis using the complete patient profile [10]. Diabetes is a chronic disease, often with comorbidities at diagnosis, so the process of diagnosis and management can include a mixture of experts from other fields, such as hepatology, nephrology, and cardiology. Opinions play a vital role in this regard where the patient's data are distributed in different hospitals, which can contribute to the decision-making process. In addition, all past studies and "No Free Lunch" theorems show that no single classifier can be considered optimal for all problems [8,10,13]. Therefore, it is hard to find a suitable single classifier. Moreover, a model generated for one community may not apply to another [14]. Many studies have developed classification models using a risk-scoring system [15]. However, there is no preferred DM risk score model. This is because the context of use, the statistical properties, the trade-off between sensitivity and specificity, and the availability of data to determine the type of used models. In addition, the false positive and false negative rates of many models raise questions about their applicability in clinical practice [16].

An ensemble of classifiers can effectively improve classification accuracy [8,17]. An ensemble method combines single classifier results and produces better performance than every single model [18]. Bagging, boosting, and stacking are the most common ensemble techniques [19]. Dietterich [20] discussed the primary motivations for combining classifiers. The goal of this work is to employ a multiple classifier system (MCS), or an ensemble classifier, to develop a prediction model to improve the accuracy of DM detection. To achieve these goals, we vertically divided a high-dimensional dataset of diabetes profiles into different sets according to medical expert opinions, diabetes clinical practice guidelines (CPGs), and correlation techniques. We carefully followed feature engineering by using representative features. Then, we trained multiple popular, diverse, and independent machine learning models based on constructed features. The algorithms are both linear (logistic regression (LR)) and nonlinear—k-nearest neighbors (KNN), naïve Bayes (NB), fuzzy decision tree (FDT), artificial neural network (ANN), decision tree (DT), and support vector machine (SVM). This means that misclassifications do not coincide. The classifiers with the best performance for each sub-dataset are combined in the proposed classification framework. This empirical evaluation of the paper is based on a real dataset with a complete set of patient description features collected from the EHR system of Mansura University Hospitals, Mansura, Egypt. The main contributions of this paper are summarized as follows:

1. An efficient ensemble of heterogeneous classifiers is proposed based on extensive evaluations. This ensemble comprises seven of the well-known techniques: KNN, NB, FDT, ANN, SVM, LR, and DT. A set of preprocessing steps is performed to enhance the quality of the sub-datasets, including feature selection, missing value imputation, normalization, codification, and discretization. The framework was applied to DM diagnosis.

2. The proposed framework used different base classifiers with varying lists of features. Each classifier has been evaluated with every sub-dataset and with different feature selection technique. The best algorithm is selected for every sub-dataset according to its performance.
3. The ensemble framework uses a combination of bagging and random subspace techniques, with a weighted voting scheme based on F-measure other than accuracy, to prevent possibly biased results.
4. The proposed classifier was evaluated by comparing its results with state-of-the-art individual and ensemble classifiers to prove its superiority.

The rest of the paper is organized as follows: Section 2 discusses the current related work. Section 3 presents the dataset used and the algorithms. Section 4 describes the proposed heterogeneous ensemble framework. Section 5 represents the results and a discussion. Finally, the conclusions and future work are summarized in Section 6.

2. Related Work

Diagnosis of DM has been extensively studied under many data mining techniques [21–23]. The most suitable data mining subfield for disease diagnosis is classification [2]. A plethora of techniques has been applied to data analytics in medical diagnosis, including single and ensemble classifiers [2,8,24,25].

2.1. Single Classifiers

Many studies in the literature used single classifier techniques for diabetes [26–32]. For example, Kavakiotis et al. [9] conducted a review of the data mining applications for diabetes. Patil et al. [21] proposed a hybrid model to diagnose type 2 DM by using two algorithms, which are simple K-means clustering to select class labels, and C4.5 to construct the classifier. The Pima Indians diabetes dataset (PIDD) from the University of California, Irvine was used to building a model with 92.38% of accuracy. Upadhyaya et al. [2] compared LR and ANN classifiers in diabetes identification problems. However, simple statistical techniques like LR could not explain the complex relationship between the utilized features and DM [24]. Sanakal and Jayakumari [22] designed a DM diagnosis model using nine features and 768 cases. The model employed a fuzzy C-means clustering algorithm and provided 94.3% of accuracy. Rahman and Afroz [23] conducted a comparative study of DM techniques for diabetes diagnosis, especially classification and clustering. Many tests were performed to measure the performance of these algorithms. The results showed that the best algorithm was the J48 classifier in the Waikato Environment for Knowledge Analysis (Weka) with 81.33% of accuracy. Varma et al. [33] proposed a diabetes diagnosis by using DT technique based on fuzzy decision boundaries, which achieved an accuracy of 75.8%. Polat et al. [34] suggested a diabetes classification system using Generalized Discriminant Analysis and a least squares SVM, which reached an accuracy of 82.50%. Beloufa and Chikh [35] proposed a fuzzy classifier using a modified artificial bee colony (ABC) optimization technique to generate fuzzy rules for DM diagnosis that achieved an accuracy of 82.68%. Chikh et al. [36] proposed a modified artificial immune recognition system by utilizing a fuzzy KNN technique. Many rule-based classifiers have been proposed [37]. However, these algorithms failed to produce balanced, optimal, and comprehensive rules [38]. Also, these algorithms were unable to provide high prediction accuracy while balancing both sensitivity and specificity. In most cases, single classifiers did not produce good performance. Their combination in an ensemble is likely to provide better prediction by forming a pool of several classifiers [19,29].

2.2. Ensembles of Multiple Classifiers

Many studies asserted that classifier ensembles offer improved performance, compared with single classifiers [26,39,40]. In addition, they can counteract choosing the worst classifier, especially with a small training dataset. The ensemble of classifiers has been used for the DM domain [8,13,26,41–43].

For instance, Zolfaghari [26] stacked an ANN and SVM using PIDD dataset. The accuracy was 88.04%, which was better than the results from single classifiers. Junior et al. [44] proposed a data stream ensemble classifier named Iterative Boosting Streaming ensemble (IBS), able to cope with classification tasks in streaming data environments. Saleh et al. [45] proposed ensemble classifier for diabetic retinopathy (DR) detection. They used fuzzy random forests (FRF) and dominance-based rough set on SRJUH dataset. They achieved an accuracy of 77%. Nanni et al. [46] proposed an ensemble of SVM base classifiers for diagnosis of Alzheimer's disease. Nguyen et al. [47] heterogeneous ensemble classifier combined with a fuzzy IF-THEN rule inference engine to capture the uncertainty in the outputs of the base classifiers. Tama and Fitri [39] utilized the AdaBoost.M1 algorithm [48] to combine SVM, C4.5, and NB. The experimental results showed that the accuracy of the SVM classifier is at the top, followed by boosting. Ali et al. [40] utilized an ensemble of AdaBoost.M1 with a random committee for DM diagnosis. The accuracy was 81% using a dataset of 18 attributes and 100 records from a local hospital. Zhu et al. [13] proposed an improved DM diagnosis method by using an MCS based on a dynamic weighted voting scheme, referred to as multiple factors weighted combination (MFWC). The authors used two datasets, which are RSMH and PIDD, to compare the MCS with five classification algorithms (SVM, NB, C4.5, LR, and ANN). MFWC outperformed all methods on both datasets. Bashir et al. [42] proposed an HMV system, a three-layer ensemble framework based on a majority voting technique, which avoided biased results due to unbalanced classes that commonly exist in DM datasets. The technique was evaluated on two datasets, which are PIDD and the Biostat Diabetes Dataset (BDD), and yielded accuracies of 93% and 77.08%, respectively. Bashir and colleagues [8] proposed HM-BagMoov, enhanced bagging, and optimized weighting algorithm. HM-BagMoov reached 77.21% and 93.07% accuracies for PIDD and BDD, respectively. The authors claimed that these results were the highest for both datasets when compared with the state-of-the-art techniques. El-Baz et al. [43] proposed two ensemble classifiers using ANN. These frameworks relied upon two base classifiers: multilayer perceptron (MLP) and a cascade-forward back-propagation network (CFBN). The first ensemble used 16 different MLPs with each base classifier having a different number of hidden neurons and a varied number of training epochs. Majority voting was used to combine the final class prediction of each classifier. The proposed classifier yielded an accuracy of 95.31% with PIDD. The second ensemble was constructed using identical settings, but CFBN was employed as a base classifier, which achieved an accuracy of 96.88%.

This literature review indicates that DM classification models are a suitable alternative to traditional clinical diagnosis. According to a literature review done by Kavakiotis et al. [9], the diabetes diagnosis problem needs further in-depth exploration. Individual classifiers can provide better performance by combining them in ensembles. However, all of the above ensemble studies have limitations. All the existing studies selected sets of base classifiers but did not discuss why these specific classifiers were selected. Recently, Tama and Rhee [41] proposed an ensemble learning technique for diabetes detection by using eight decision tree classifiers. The authors failed to clarify the reasons for selecting a decision tree over other classifiers. Most of the previous studies used homogeneous ensembles. To the best of our knowledge, no study builds a classification system by combining the results of the most accurate heterogeneous techniques to diagnose DM. In other words, for high-dimensional data, there are no studies that can partition data vertically using some technique (such as correlation into different subsets) that evaluates the most popular classifiers on these different subsets and that builds an ensemble using the most accurate algorithms as base classifiers for each specific subset. In addition, most DM studies in the data-mining field depend on public datasets, such as PIDD [49], which has no representative feature sets. DM is a chronic disease. At the time of diagnosis, the patient could have other complications in the heart, kidney, liver, etc. Collecting all these features provides a complete picture of the patient. However, they cannot be used with a single classifier. All studies depended on a small number of classifiers in their ensembles. Most studies did not examine the performance differences between the ensemble classifier and its base classifiers. Creating a powerful ensemble requires many diverse base classifiers, different training subsets, and various feature sets.

To overcome the aforementioned limitations of diabetes diagnosis classifiers, this study proposes a novel ensemble classifier. The novelty of this framework can be summarized in a set of points. First, the study is based on a real dataset with a complete set of diabetes patients' characteristics. This dataset is medically divided into different feature sets, such as symptoms and others. Second, the framework is based on a set of seven heterogeneous classifiers from different domains, such as statistical, structural, probabilistic, fuzzy, and logical. Third, feature selection is based on two different techniques to select the best feature set for every classifier. These base classifiers were evaluated on every subset, and the best algorithm was selected for every dataset. Finally, these algorithms formed the ensemble, and we used a weighted voting function to get the final decision based on the base classifiers' F-measure. The proposed approach is medically more intuitive, and it can be used to design diagnosis systems for any other diseases. It takes a decision based on the medical importance level of different types of data. Our technique handles the already existing problems in medical data, such as missing values, a large number of features, and various data formats.

3. Materials and Methods

3.1. Dataset Description

The dataset was obtained from the hospitals of Mansoura University, Mansoura, Egypt, for the period between January 2010 and August 2013. Domain experts collected all the features that can add value in diabetes diagnosis. Sixty-seven patients were enrolled in this study, but seven control subjects were excluded due to limited blood samples. Table 1 shows descriptions of features that are considered in this study.

Table 1. Dataset descriptions, where data type is {N = Numerical, C = Categorical}.

Feature Type	Feature Name	Data Type	Normal Range	UoM	Min-Mean-Max	Feature No.
Demographics	Residence	C	{Urban, Rural}	-	-	1
	Occupation	C	{NHW, HW, Non}	-	-	2
	Gender	C	{Male, Female}	-	-	3
	Age	N	20–80	year	29–48–74	4
	BMI	N	18.5–25	kg/m^2	20–33.117–45	5
Sugar lab tests	HbA1C	N	≤ 5.7	%	5–6.373–7.4	6
	2h PG	N	≤ 139	mg/dl	165–202.733–235	7
	FPG	N	≤ 99	mg/dl	96–129.633–156	8
Hematological profile	Prothrombin INR	N	0–1	%	1–1.16–1.4	9
	Red cell count	N	4.2–5.4	10^6/cmm	3.8–5.194–5.88	10
	Hbg	N	12–16	g/dL	9.8-12.332-13.4	11
	Hematocrit (PCV)	N	37–47	vol%	31.1–35.215–36.8	12
	MCV	N	80–90	fl	26.8–71.908–76.4	13
	MCH	N	27–32	pg	3.3–25.47–29.4	14
	MCHC	N	30–37	%	1.8–35.465–41.7	15
	Platelet count	N	150–400	10^3/cmm	135–316.183–2000	16
	White cell count	N	4–11	10^3/cmm	6–8.055–9.2	17
	Basophils	N	0–1	%	0–1.013–5	18
	Lymphocytes	N	20–45	%	21.2–25.768–29	19
	Monocytes	N	2–10	%	1.7–2.942–4	20
	Eosinophils	N	1–4	%	1–1.897–3.4	21

Table 1. *Cont.*

Feature Type	Feature Name		Data Type	Normal Range	UoM	Min-Mean-Max	Feature No.
Symptoms	Urination frequency		C	{normal, +, ++, +++}	-	-	22
	Vision		C	{normal, +, ++, +++}	-	-	23
	Thirst		C	{normal, +, ++, +++}	-	-	24
	Hunger		C	{normal, +, ++, +++}	-	-	25
	Fatigue		C	{normal, +, ++, +++}	-	-	26
Kidney function lab tests	Serum potassium		N	3.5–5.3	mEq/L	2.4–3.767–4.3	27
	Serum urea		N	5–50	mg/dL	17–31.56–67	28
	Serum uric acid		N	3.0–7.0	mg/dL	3–4.237–7.9	29
	Serum creatinine		N	0.7–1.4	mg/dL	0.9–1.35–3.6	30
	Serum sodium		N	135–150	mEq/L	134–137.833–158	31
Lipid profile	LDL cholesterol		N	0–130	mg/dL	50–94.917–170	32
	Total cholesterol		N	0–200	mg/dL	158–209.367–275	33
	Triglycerides		N	60–160	mg/dL	78–144.767–189	34
	HDL cholesterol		N	45–65	mg/dL	30–55.533–65	35
Tumor markers	Ferritin		C	28–397	ng/mL	-	36
	AFP serum		C	0.5–5.5	IU/ml	-	37
	CA-125		C	1.9–16.3	U/mL	-	38
Urine analysis	Chemical examination	Protein	C	{normal, +, ++, +++}	-	-	39
		Blood	C	{normal, +, ++, +++}	-	-	40
		Bilirubin	C	{normal, +, ++, +++}	-	-	41
		Glucose	C	{normal, +, ++, +++}	-	-	42
		Ketones	C	{normal, +, ++, +++}	-	-	43
		Uro-bilinogen	C	{normal, +, ++, +++}	-	-	44
	Microscopic examination	Pus	C	{normal, +, ++, +++}	-	-	45
		RBCs	C	{normal, +, ++, +++}	-	-	46
		Crystals	C	{normal, +, ++, +++}	-	-	47
Liver function tests	S. albumin		N	3.5–5.0	g/dL	1.9–4.082–5.4	48
	Total bilirubin		N	0.0–1.0	mg/dL	0.8–1.317–3	49
	Direct bilirubin		N	0.0–0.3	mg/dL	0.3–0.533–1.6	50
	SGOT (AST)		N	0–40	U/L	35–54.567–165	51
	SGPT (ALT)		N	0–45	U/L	35–57.317–183	52
	Alk. phosphatase		N	64–306	U/L	170–214.2–360	53
	γ GT		N	7–32	U/L	18–35.833–98	54
	Total protein		N	6.0–8.7	g/dL	3.1–4.858–8.7	55
Diseases	Patient disease		C	{yes, no}	-	Collection of diseases	59
Diagnosis	Diabetes diagnosis		C	{diabetes, no diabetes}	-	-	60

The independent or input variables are a list of 60 integrated patient characteristics, which are five features of patient demographics, three features of sugar level tests, 13 features of hematological profiles, five features of symptoms, five features of kidney function lab tests, five features of lipid profiles, three features of tumor markers, nine features of urine analysis, eight features of liver function lab tests, three features of female histories, and one feature for complications. Because DM is a chronic disease with many probable complications at diagnosis, these features provide a complete picture of the patient history and support the making of an accurate decision. A dependent variable (target, class, or output variable) is a binary variable with two categories: 0 means no diabetes and 1 indicates diabetes. The dataset was distributed into 53% (cases with diabetes) and 47% (controls). The dataset is

balanced because the class feature divides the dataset approximately in half. Some features, such as patient diseases, require a unification of terminology for medical terms. We used the Systematized Nomenclature of Medicine—Clinical Terms (SNOMED CT) standard terminology to standardize and unify these terms [50].

3.2. Base Classifier Algorithms

The proposed framework utilizes seven popular classification algorithms, which are DT, KNN, SVM, NB, ANN, LR, and FDT. The selection was based on their ability to predict categorical features, their research streams, and diversity (e.g., statistical, structural, probabilistic, fuzzy, or logical) [51]. These techniques have been used individually in many diabetes studies [4,14,49]. This selection helps to reduce model bias and supports the comparative assessments of model performance. To attain diversity in our ensemble model, these algorithms are entirely different. The classification process can be defined as follows. Given a $D = n \times d$ training dataset, and a class label value y_k in $v = \{y_1, \ldots, y_k\}$ associated with each of the n cases in D, i.e., $D = \{\{X_1, y_1\}, \{X_2, y_2\}, \ldots, \{X_n, y_k\}\}$, and given that X_i represents the d-dimensional tuples associated with classes y_i. It creates a training model M able to predict the class label of a d-dimensional record $\bar{Y} \notin D$. Mathematically, classifier M can be defined as a function (f), which takes a case in the d dimensional search space $\overline{X} \notin D$ and assigns it a label value $\overline{y}$; $M : f\left(\overline{X}\right) \rightarrow \overline{y}$, where $\overline{y} \in \{y_1, y_2, \ldots, y_n\}$. The following subsections provide a brief discussion for the utilized classifiers.

3.2.1. Decision Tree

DT is popular in the medical domain as a powerful classification algorithm [10]. A DT produces a transparent tree structure that allows the decision maker to check and interpret the resulting model. The DT can work with a large volume of data, and handle both continuous and categorical features. The Iterative Dichotomiser 3 (ID3), C4.5, C5.0, classification and regression trees (CART), and chi-squared automatic interaction detector (CHAID) are the most common DT algorithms. This paper is based on the C4.5 algorithm. Tree building starts at the root node with the entire dataset split in a top-down approach using the most suitable feature. This feature is removed from the splits followed by recursive partitioning of the splits into smaller subsets. The feature that best partitions the samples into distinct classes is based on specific measures, such as information gain, gain ratio, and Gini index. Our study uses the most popular technique for information gain (Equation (1)), which is based on the level of impurity or entropy. For feature A and a collection of examples S:

$$Information\ Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (1)$$

where $Value(A)$ is the set of all possible values for attribute A, $S_v = \{s \in S | A\{s\} = v\}$, and $Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i$, where c is the number of classes and p_i is the proportion of S belonging to class i. At each node, the DT chooses the feature with the highest information gain in order to split the dataset. To avoid overfitting, the generated tree can be pruned to remove non-essential terminal branches without affecting classification accuracy. The overall computational complexity of this algorithm is $O_{DT}\left(mn^2\right)$, for n is the number of instances, and m is the number of features.

3.2.2. Support Vector Machine

SVM nonlinearly maps the training data to a higher dimensional space. It separates the different classes of data by defining a separating hyperplane, i.e., a decision boundary. It has a good generalization ability, robustness for high-dimensional data, and better performance than ANNs, especially for binary classification [52]. On the other hand, SVM is very sensitive to uncertainties, and the high-dimensional

space can lead to overfitting. SVM defines the hyperplane by using support vectors (training tuples on the plane) and margins (represented by the support vectors), as shown in Figure 1.

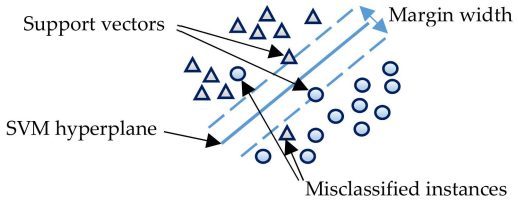


Figure 1. The SVM classification with a hyperplane.

SVMs try to minimize classification errors by maximizing the margin between the separating hyperplane and the datasets. A separating hyperplane can be written as:

$$W \times X + b = 0 \tag{2}$$

where $W = \{w_1, w_2, \ldots, w_n\}$ is a weight vector (n is the attribute number), and b is bias. For a dataset of two features, i.e., $X = (x_1, x_2)$ and $b = w_0$, the hyperplanes in the figure define the margin based on support vectors, and can be written mathematically as seen in Equations (3) and (4):

$$H_1 : w_0 + w_1 x_1 + w_2 x_2 \geq 1 for\ y_i = +1 \tag{3}$$

$$H_2 : w_0 + w_1 x_1 + w_2 x_2 \leq 1 for\ y_i = -1 \tag{4}$$

Any case that falls on or above H_1 belongs to class +1, and any that fall on or below H_2 belong to class −1. The overall computational complexity of this algorithm is $O_{SVM}\left(n^3\right)$, for n is the number of instances.

3.2.3. Naïve Bayes

NB is a statistical classifier based on Bayes' theorem. It is based on the class conditional–independence assumption, where the effect of an attribute value on a given class is independent of the values of the other attributes. The NB technique operates as follows:

1. For training set D of cases and their associated class labels, each case is represented by a vector of n-dimensional attributes, $X = (x_1, x_2, \ldots, x_n)$ for n values of n features $(A_1, A_2, \ldots, A_n)$. Each case can be classified as one of the m classes: $(C_i, C_2, \ldots, C_m)$.
2. For a new case, X, NB predicts that X has the class having the highest a posteriori probability, conditioned on X. In other words, the NB classifier predicts that case X belongs to a class C_i if and only if $P(C_i|X)$ in Equation (5) is the largest, and C_i is the maximum a posteriori hypothesis:

$$P(C_i|X) > P\left(C_j\middle|X\right) for\ 1 \leq j \leq m, j \neq i \tag{5}$$

Based on Bayes' theorem, $P(C_i|X)$ is calculated with Equation (6):

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{6}$$

3. Only $P(X|C_i)$ needs to be optimized or maximized because $P(X)$ has the same value for all classes, and if the class prior probabilities are not known, then, it is usually assumed that all classes have the same probability value, $P(C_1) = P(C_2) = \ldots = P(C_m)$.

4. Datasets are usually of multiple attributes, so it would be computationally extremely expensive to compute $P(X|C_i)$. Using the naive assumption of class conditional independence, $P(X|C_i)$ is calculated with Equation (7), and $P(x_k|C_i)$ is calculated according to the type of the feature:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i) \quad (7)$$

The overall computational complexity of this algorithm is $O_{NB}(mn)$, for n is the number of instances, and m is the number of features.

3.2.4. Artificial Neural Network

An ANN is a mathematical formulation of the human neural architecture. It is organized in layers with one input layer, one or more hidden layers, and one output layer. Neurons in one layer are connected with each neuron in the next layer by weighted connections. The weight value w_{ij} is the strength of the link between the i-th neuron in a layer and the j-th neuron in the next layer. The complexity of the model determines the number of layers and the number of neurons in each layer. A general scheme for a three-layer network is shown in Figure 2.

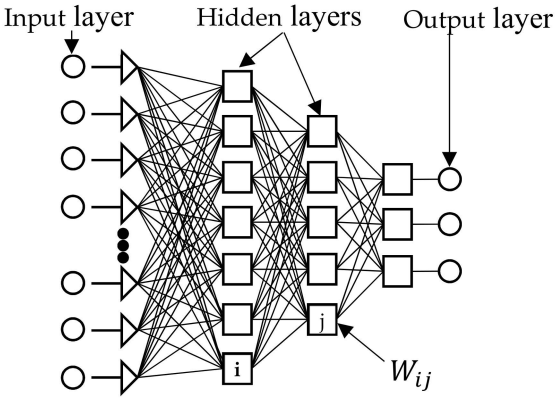


Figure 2. Illustration of an MLP network.

The input layer's neurons receive the input data (activation values) and pass them to the first hidden layer's neurons via weighted connections. These data are mathematically processed, and the results are transferred to the neurons in the next layer. The network's output is generated from the neurons in the last layer. Neuron j in a hidden layer processes the incoming data (x_i) in three steps:

(1) Calculate the weighted sum and add a bias term (θ_j) according to Equation (8):

$$val_j = \sum_{i=1}^{m} x_i \times w_{ij} + \theta_j \ (j = 1, 2, \ldots, n) \quad (8)$$

(2) Transform val_j through a suitable mathematical transfer function, such as unit step (threshold), piecewise linear and Gaussian sigmoid, or sigmoid (given in Equation (9); and
(3) Transfer the result to neurons in the next layer until it reaches the output nodes (feed-forward):

$$f(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

The difference between predicted value and actual value (error) is propagated backward by apportioning it to each node's weight to modify it (feed-backward). This training process loops until

the ANN reaches a state of equilibrium. For the final user, the network is a "black box" that receives an input vector with m values and provides an output vector with n results. The learning process from a series of examples is achieved by representing each case as the input vector $X_{im} = (x_{i1}, x_{i2}, \ldots, x_{im})$ and output vector $Y_{in} = (y_{i1}, y_{i2}, \ldots, y_{in})$. The training process tries to approximate function f between the vectors X_{im} and Y_{in}, i.e., $Y_{in} = f(X_{in})$. This objective is reached by iteratively changing the values of the connection weights (w_{ij}) according to a suitable mathematical rule. More details about ANN were provided by Basheer and Hajmeer [53]. The overall computational complexity of this algorithm is $O_{ANN}(emnk)$, for n instances, m features, e epochs, and k neurons.

3.2.5. Logistic Regression

LR is a statistical technique that is a generalization of linear regression. It has two main types, binary LR (BLG), used for the binary dependent variable (i.e., the outcome is "0" or "1".), and multinomial LR (MLR), used for a dependent variable with more than two categories. When working with LR, we need to make an algebraic conversion to arrive at our usual linear regression equation, $Y = \beta_0 + \beta_1 X + e$. BLG estimates the probability of a binary response based on a set of predictor (independent) variables that may be continuous, discrete, dichotomous, or a mix of any of these. The BLR curve is constructed using the natural logarithm of the odds of the target variable. The odds are the probability that a particular outcome is that of a case divided by the probability that it is a noncase (i.e., $Ln\frac{p}{1-p}$). The logistic (logit) transformation is the logarithm of the odds of the positive response and is defined in Equation (10):

$$\eta_i = Ln\frac{p(x)}{1-p(x)} = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{10}$$

where $X = [x_1, x_2, \ldots, x_n]^T$ is the set of predictor variables, and $\beta = [\beta_1, \beta_2, \ldots, \beta_n]^T$ is the set of regression coefficients. Solving for p is done with Equation (11):

$$p = \frac{e^{(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}}{1+e^{(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}} \tag{11}$$

where β_0 is a constant that moves the curve left and right, and β_i is the slope that defines the steepness of the curve, for $i = 1, 2, \ldots, n$, n is the number of predictors. It uses maximum likelihood estimation (MLE) to obtain the model coefficients that relate predictors to the target, as shown in Equation (12):

$$\beta^1 = \beta^0 + \left[X^T W X\right]^{-1}.X^T(y-\mu) \tag{12}$$

where β is a vector of the LR coefficients, W is a square matrix of order N with elements $n_i\pi_i(1-\pi_i)$ on the diagonals and zeros everywhere else, and μ is a vector of length N with elements $\mu = n_i\pi_i$. After the estimation of this initial function, the process is repeated until the log likelihood (LL) does not change significantly. A pseudo R^2 value (e.g., Efron's, McFadden's, and Count) is used to indicate the adequacy (goodness-of-fit) of the regression model. The overall computational complexity of this algorithm is $O_{LR}\left(\mathrm{nm}^2\right)$, for n instances and m features.

3.2.6. Fuzzy Decision Tree

The medical domain is usually imprecise in nature. Handling the fuzziness of data in a classifier is critical. This study uses the fuzzy C4.5 algorithm, which improves on the performance of C4.5. The overall computational complexity of this algorithm is O_{FDT}, and it is equal to the complexity of DT. Using CPGs and domain experts, we first formulated the fuzzy sets for all of the used numerical features. Secondly, we fuzzified the preprocessed training datasets with linguistic labels of fuzzy sets that have the highest compatibility with the input values.

More formally, for k samples, crisp dataset D represented by n features, $[F_1, F_1, \ldots, F_n]$, the n-dimensional tuple $T_i = [a_1, a_1, \ldots, a_n]$ is represented as a kn-dimensional vector: $T_i = \left[< \mu_{FT_1}[a_1], \mu_{FT_2}[a_1], \ldots, \mu_{FT_k}[a_1] >, \ldots, < \mu_{FT_1}[a_n], \mu_{FT_2}[a_n], \ldots, \mu_{FT_k}[a_n] > \right]$ where $\mu_{FT_k}(a_i)$ represents the degree of membership of the fuzzy term FT_k of feature F_i ($F_i = a_i$), k is the number of terms, and n is the number of variables. D is converted to fuzzy dataset D^F. If linguistic variable F_n has k fuzzy terms, $FT_1, FT_2, \ldots, FT_k$, then for each crisp value v of F_n, the representative fuzzy value is $\max\left\{\mu_{FT_1}\{v\}, \mu_{FT_2}\{v\}, \ldots, \mu_{FT_k}\{v\}\right\}$, or $\mu_{FT_j}(v) \geq 0.5$. For example, if serum uric acid = 3.4, and its fuzzification is $\mu_{Low}(3.4) = 0.44$, $\mu_{Normal}(3.4) = 0.56$, and $\mu_{High}(3.4) = 0.00$, then the selected label for this value is Normal. We performed some preprocessing for the generated discretized data by removing the redundant vectors (cases). Finally, we created the FDT by applying the C4.5 algorithm to the resulting fuzzy training sets using Weka's J48 algorithm.

3.2.7. K-Nearest Neighbors

These kind of algorithms are distance-based classifiers that do not explicitly build models. The class value of a new case is equal to the class of its nearest neighbor, based on a specific distance equation. Heterogeneous Euclidean-Overlap Metric (HEOM) can be used for the distance measure to determine the K-nearest neighbors. HEOM calculates different distance measures for different types of attribute. Euclidean distance is used for numerical features with Equation (13):

$$D_N\left(X^i, X^k\right) = \sqrt{\sum_j \left(x_j^i - x_j^k\right)^2} \tag{13}$$

where X^i and X^k are two cases, x_j^i and x_j^k are the j feature in both cases ($j = 1, 2, \ldots n$), and N is the number of features. Categorical features use the binary equation in Equation (14):

$$\left(X^i, X^k\right) = \sum_j d_j\left(x_{aj}, x_{bj}\right), d_j\left(x_{aj}, x_{bj}\right) = \begin{cases} 0 x_{aj} \neq x_{bj} \\ 1 x_{aj} = x_{bj} \end{cases} \tag{14}$$

For input case x, the KNN technique selects the k nearest neighbors and represents it in $V_x = \{V_k\}_{k=1}^K$, for V_k as the k nearest neighbor; and the output equals the output of the majority of these samples. If cases contain both numerical and categorical features, then the total distance is $D_T\left(X^i, X^k\right) = \sum_{q \in Q} D_N + \sum_{c \in C} D_C$ for q numeric and c categorical features, and $q + c = n$. An appropriate choice for k is very important, such as k= 3 to select the nearest three cases. Once the nearest neighbor list is selected, the new case can be classified based on a voting method, such as majority voting or distance weighted voting. In majority voting, the total vote $T_i(t)$ of the neighbors of X^i having the label t is $T_i(t) = \sum_{k \in V_x}(I(t, y_k))$, where $I(t, y_k) = 1$ if $t = y_k$, and $I(t, y_k) = 0$ otherwise. The overall computational complexity of this algorithm is $O_{KNN}(n\ logk)$, for n instances and k neighbors.

3.3. Classifier Ensembles

A classifier ensemble, or a meta-classifier, is the combination of different models to produce a stronger and stable one. There are many classifier ensemble techniques, including bagging (i.e., bootstrap aggregation), boosting, stacking, random subspace, decorate, and rotation forest [54–58]. They can increase the predictive performance of a single model. A detailed discussion of these techniques was provided by Kuncheva [19]. This study uses a combination of random subspace (RS) [54] and bagging [55,57] techniques. RS is based on the theory of stochastic discrimination. It projects different feature vectors, v_i, into fewer-dimensional subspaces, without replacement, in order to train ensemble members m_i. This technique is suitable for medical applications that have highly dimensional data. The key issue of how to select v_i is solved by collecting the features that are medically related, such as liver tests, kidney tests, glucose level tests, and symptoms. In addition, the medically collected features are correlated. The weighted voting techniques are used from the bagging

method. To calculate the final decision, the votes are multiplied by weights obtained from the classifier performance metrics (such as accuracy) with $w_i = \log\left(\frac{p_i}{1-p_i}\right)$, where p_i is the accuracy of the i^{th} classifier. To improve the performance in this study, weights are based on F-measure.

4. The Proposed Diabetes Ensemble Classifier

The combination of outputs from several different models is an obvious approach to making decisions that are more reliable. In data mining, this is called ensemble classifiers. Our model works like a committee of experts, where each expert is a classifier. Each expert is specialized in a limited domain. The committee often comes up with a wiser decision than individual experts do. The opinions of all experts (i.e., the classifiers) are amalgamated for consideration by using any mechanism, such as weighted voting. An ensemble classifier is seldom less accurate than individual classifiers, but errors still occur, because no training scheme is perfect [59]. Errors depend on how well the algorithm matches the problem at hand and the quality of the training data (i.e., data preprocessing). To enhance this process, we tested seven well-known classifiers with every preprocessed sub-dataset, selected the classifier with the best performance for each sub-dataset, and collected their F-measures. The final output is based on a weighted voting technique. The proposed framework involves domain and data understanding, data preprocessing, data distribution, and ensemble building. Figure 3 shows the detailed architecture of the proposed ensemble framework.

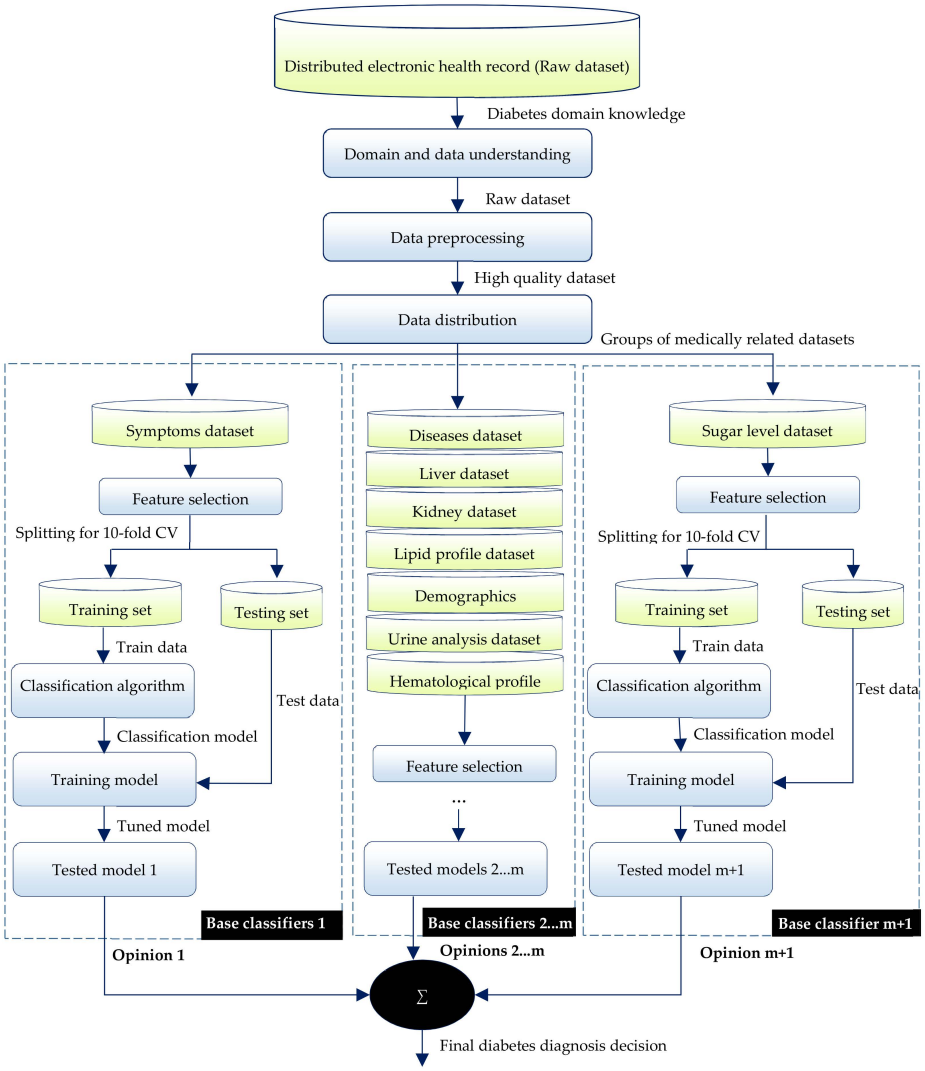


Figure 3. The detailed architecture of the proposed ensemble framework.

4.1. Domain and Data Understanding

This step is critical to understand the nature of diabetes, its critical characteristics, and the right diagnosis process. Domain experts participated in this process with the help of some of the most recent diabetes CPGs [60,61]. In another study, authors created a standard diabetes diagnosis ontology in Web Ontology Language 2 (OWL 2) format, which deeply studies this issue [50]. According to the most recent CPGs, a diabetes diagnosis cannot be made by only conducting lab tests for glucose levels. All of the patient profile is critical to making the right decision. In this study, we collect these complete sets of patient characteristics.

4.2. Data Preprocessing

Data preprocessing tasks are necessary to transform the original raw information with incomplete, inconsistent, and noisy data into a high-quality and cleaned dataset for subsequent analysis. The classification performance can be improved mainly by selecting the right combination of preprocessing methods [62]. There is no predefined sequence of preparation steps. We used Weka 3.8.1 application programming interface (API) to finish this step. The major tasks are in the following sequence.

Step 1: Unified unit of Measurement

All numerical features are lab tests with different units of measurement (UoMs). The raw dataset has many features with many units of measurement. For example, the two hour plasma glucose (2h PG) feature has some values in millimoles per liter (mmol/L) and some in milligrams per decaliter (mg/dL). This produces an inconsistent dataset, e.g., 11.1 mmol/L = 200 mg/dL. All features are converted to use unified UoMs.

Step 2: Missing Value Imputation

In our dataset, the class label feature has 0% missing values, and there are no cases with a large number of missing values, so no cases are entirely deleted. We have some features with a large percentage of missing values, such as CA-125, α-fetoprotein (AFP) serum, and ferritin. These features are removed from the dataset. The remaining attribute set has 57 features. All other features have 0% missing values.

Step 3: Outlier Detection and Prevention

Outliers and extreme values affect the performance of the classifier. We used interquartile range as a filter for detecting outliers and extreme values. The platelet count feature has outliers in four cases (where the value is 2000), but the most abnormal value could be 400. This value is replaced by the average of this feature, which is 195.91.

Step 4: Data Normalization, Transformation, and Coding

The normalization process has many techniques, such as z-score and min-max. In this model, all numerical features are rescaled into the interval $[0, 1]$ to have the same effect in the classification algorithm. We used the min-max technique. Equation (15) gives a general formula to normalize A in a specific [C, D] range, where A is the old value and B is the normalized value, and the range used in our case is [0.0, 1.0]:

$$B = \left(\frac{A - minimum\ value\ of\ A}{maximum\ value\ of\ A - minimum\ value\ of\ A} \right) \times (D - C) + C \tag{15}$$

The raw dataset has some features that are transformed into other meaningful ones. For example, weight in kilograms and height in meters are transformed to body mass index (BMI) in kg/m^2 as follows: $BMI = weight\ (kg)/(height(m))^2$. Medical data need some form for the unification of the contents.

The occupation feature has many jobs, so we convert its values into "not hard work", "hard work" and "non." Many other categorical features, such as vision and frequency of urination, have many inconsistent values. With the guidance of a domain expert, we encode these values in a unified manner. As another example, the raw medical data for frequency of urination are 3–5 times, 6–8 times, 9–10 times, and more than 10 times, encoded to normal, +, ++, +++, respectively.

Step 5: Discretization

This process is performed on the numerical features to partition values into a finite number of non-overlapping intervals. Finding the optimal discretization of a feature is NP-hard [62]. There are two main techniques of discretization, namely supervised method, where the class feature is considered, and the unsupervised method, where the class feature is not considered. In methods such as equal width and equal frequency, a predefined number of bins (n) is determined. Because defining the optimal number of bins in unsupervised methods is complex, we utilized the supervised method based on Fayyad and Irani's MDL method [63].

4.3. Data Distribution

The main dataset is divided into different complementary subsets. Each subset is represented by a smaller number of features (nine groups), as shown in Table 1. Building an ensemble classifier's base models with different sets of features can be done randomly, where a set of N features can be randomly distributed to M models [64]. A more intuitive way is to distribute these features according to their medical and algorithmic correlations. According to domain expert opinions and diabetes CPGs [60,61], the set of features is divided into 10 subsets. One of these sets is removed in the preparation step because it has many missing values. Each set contains a medically related set of features. We used a correlation technique to recheck the association of these features. Each group is used with a specific base classifier, all of which are collected in the combined ensemble framework.

4.4. Building the Ensemble Classifier

In this section, we discuss the construction of the complete ensemble classifier. To achieve this goal, we have to select the best classifier for each dataset with the most suitable feature set. The overall process is formulated in Algorithm 1. This phase has two main steps that are discussed in this section.

4.4.1. Feature Selection

Even the best classifiers perform poorly if the set of features is not chosen correctly. As a result, feature selection (FS) is one of the most critical factors for building efficient classifiers. FS improves the prediction performance, avoids overfitting, and provides faster and more cost-effective predictors. There is no perfect FS technique for all datasets, and the selection is based on the evaluation process. FS techniques can be a model-free (i.e., a filter) approach, which selects features independently of a classifier based on distance, correlation, or information theoretic measures (e.g., Chi-squared, gain ratio, or information gain), or a model-based (wrapper) approach. It applies specific classifiers (e.g., DT) and uses their accuracies based on 10-fold cross validation as a measure of subset effectiveness. For each prepared dataset, a diverse combination of FS methods is utilized, including the filter method by correlation-based feature selection (CFS), and the wrapper method by using a classifier (e.g., the 1R classifier). Hall and Holmes [65] asserted that CFS and wrappers as the most suitable FS methods. The main part of CFS is heuristics to evaluate the importance or the merits of attributes to predict the label class, obtained with Equation (16):

$$A_F = \frac{\sum_j U\left(A_j, C\right)}{\sqrt{\sum_i \sum_j U\left(A_i, A_j\right)}} \tag{16}$$

where A_F is the merit of feature subset F, C is the class attribute, and the indices i and j range over all attributes in the set. First, all numerical features are discretized; the correlation between two nominal attributes, A and B, can be measured using symmetric uncertainty from Equation (17):

$$U(A,B) = 2 \times \frac{H(A) + H(B) - H(A,B)}{H(A) + H(B)} \tag{17}$$

where H is the entropy function, $H(A,B)$ is the joint entropy of A and B, and $U(A,B) \in [0,1]$. CFS's CfsSubsetEval technique uses the GreedyStepwise search method, which performs a greedy forward or backward search through the list of attribute subsets. We measured the performance of all selected classifiers with each FS technique. Based on the evaluations, we selected the best FS technique for every sub-dataset for every classifier.

Algorithm 1. Construction of an enhanced ensemble classifier.

Input:

- D: a set of $n \times d$ training tuples + class label vector $L = \{0, 1\}$ (0: no diabetes, 1: diabetes)
- M: a pool of classifiers, $M = \{DT, SVM, ANN, KNN, NB, LR, FDT\}$

Output:

- $\overline{M}$: the trained composite model
- Z: the output of the ensemble for new cases

Method:

1. $D_i \leftarrow \{(n \times r_i) \in \|D\|, \forall i \in 1, 2, \ldots t, \sum_i r_i = d\}$. // $\|D\|$ *is all vertical partitions of* D *with* r_i *attributes.* // *according to a correlation algorithm and expert opinion.*
   $s \leftarrow i$ // *the ensemble size*
2. $V \leftarrow$ base classifiers weight vectors based on their F-measures
3. ***for*** $j = 1$ to s ***do***
4. ***for*** k = 1 to $|M|$ ***do*** // $|M|$ *is the number of classifiers in* M.
5. train (M_k, D_j) // ***for*** $D_j = \left(n \times r_j\right)$, $M_k \in M$ *is a heterogeneous base classifier.*
6. test (M_k, D_j, TA) // *for* TA *is a testing method such as k-fold cross-validation.*
7. ***end for***
8. select the model M_j with the best F-measure for the set D_j
9. $V+ =$ F-measure of M_j
10. ***end for***
11. $\overline{M} \leftarrow \sum_{j=1,2,\ldots,s} M_j + V$
12. ***for*** a new unseen instance X ***do***
13. - distribute X vertically as done in step 1
14. - classify X by $\overline{M}$
15. - final decision for X is $Z \leftarrow \underset{c_j \in V}{\operatorname{argmax}} \sum_{i=1}^{M} w_{c_j}^i(X) f_{c_j}^i(X)$
16. - ***Return*** Z
17. ***end for***

4.4.2. Selecting and Building Base Classifiers

The ensemble classifier is a technique to enhance the accuracy of composite models [13]. However, without accurate and proper design, the combined model may perform worse than individual classifiers. A crucial step in the design process is to select the optimal set of base classifiers. The selection is based on the accuracy of these techniques. There are two categories of ensemble framework [17]: the homogeneous framework, which uses base classifiers of the same type, and the heterogeneous framework, which uses base classifiers of different types. The ensemble approach requires a level of

disagreement between member classifiers (model diversity) to cover errors, and this can be achieved in the heterogeneous approach [42,66]. Many studies asserted that the power of a heterogeneous ensemble has a strong relation to the performance of the base classifiers and the lack of correlation between them [8]. As a result, we used the heterogeneous approach based on an RS method. We selected seven of the best-known algorithms that produced high accuracy in the medical domain to become our base classifiers. Each classifier has a diverse set of qualities that complement each other to form an accurate ensemble model. Each classifier is trained using all training sub-datasets from the previous step and with two types of feature selection algorithms. Based on a collection of evaluation metrics, the best algorithm was selected for every sub-dataset and with a specific set of features. Building an ensemble based on different base classifiers where each one works on various feature sets can improve the performance of the combined model [64].

4.4.3. Ensemble of Base Classifiers

The most popular types of integration are algebraic methods (e.g., sum, weighted average, min, max, etc.) and voting methods, including unweighted voting (i.e., a plurality or majority) and weighted voting [67,68]. Voting methods are more accurate than algebraic ones. In unweighted voting, each model suggests a class value and, from Equation (18), the ensemble proposes the class with the most votes:

$$class(x) = argmax_{c_i \in dom(y)} \sum_k g(y_k(x), c_i), g(y, c) = \begin{cases} 1 y = c \\ 0 y \neq c \end{cases} \quad (18)$$

where $y_k(x)$ is the class result of the $k-$th classifier, and $g(y, c)$ is an indicator function. For instance, Majid et al. [69] used an IDM-PhyChm-Ens classifier based on majority voting for cancer prediction using amino acid sequences. This voting is suitable if the learning schemes perform comparably well. In the weighted voting scheme, if base classifiers produce different predictions, then the final prediction will be based on all of the classifier weights. Weights can be assigned statically or dynamically [13]. The weights can be assigned based on the classifier accuracy, where the classifier with high accuracy attains a high weight, and vice versa. The final classification is based on this objective function (OF). However, the classifier can have biased accuracy results based on a biased dataset if there are unbalanced classes. The OF should be as contradictory as possible to achieve the highest performance. In addition, we need an unbiased metric to assign the weights to the base classifiers, instead of the accuracy measure. In our framework, a multi-objective OF is used based on F-measure (i.e., a weighted average of precision and recall) calculated in the training phase of the base classifiers. If there are M base classifiers, and X is the new case to be decided, the final decision is calculated with Equation (19):

$$Z = \underset{c_j \in V}{\text{argmax}} \sum_{i=1}^{M} w_{c_j}^i(X) f_{c_j}^i(X) \quad (19)$$

where Z is the output class for X; V is the set of possible classes; $w_{c_j}^i(X)$ is the ith classifier's weight based on its F-measure; and $f_{c_j}^i(x) \in \{0, 1\}$ is the decision result of the ith classifier for X. If the ith classifier predicts that X belongs to a class c_j, then give f a value of 1; otherwise, the value is 0. We used an enhanced combination of bagging and random subspace, as shown in Figure 3. Bagging builds models using random horizontal subsets of the original training set, and then, classifies a new instance by aggregating the individual model predictions to form a final prediction. Bagging reduces overfitting and works best with strong models, such as SVM, DT, and NB. On the other hand, random subspace divides the dataset vertically into different feature sets. Each set is used with a specific classifier. For a new instance, each trained classifier predicts one class of 0 or 1, and a voting technique is used to provide the final decision. For example, suppose the trained base classifiers produce the following F-measures in the training phase: SVM = 0.6, DT = 0.3, NB = 0.9, ANN = 0.89, LR = 0.85, FDT = 0.5, and KNN = 0.35. Now, suppose the classifiers have predicted the following classes for a new test

instance: SVM = 0, DT = 0, NB = 1, ANN = 1, LR = 1, FDT = 0, and KNN = 0. The weighted vote Z is calculated as follows for each class—class 0: SVM + DT + FDT + KNN → 0.6 + 0.3 + 0.5 + 0.35 = 1.75, and class 1: NB + ANN + LR → 0.9 + 0.89 + 0.85 = 2.64. Hence, the new instance is put into class 1 because it has been classified with only three (but strong) classifiers.

5. Results and Discussion

This section discusses the evaluation process of our ensemble classifier and all of its base classifiers. As shown in Algorithm 1, many parameters need to be calculated. Each of the seven algorithms is used with each sub-dataset, and results are collected. For each algorithm, the evaluation is done using different feature sets according to different FS techniques. The purpose of these comparative evaluations is to select the best feature set for each algorithm based on the natures of the dataset and the classifier. The results of the selections are combined in the proposed ensemble classifier to take the final decision. The primary focus of this work is to show the feasibility and suitability of the data mining framework for DM diagnosis. To keep our work focused and data-efficient, we used the default Weka recommended model parameters instead of performing hyper-parameter tuning.

5.1. Evaluation Metrics

To calculate the performance efficiency of our ensemble framework, a set of 11 metrics was used, including F-measure and accuracy. In this study, diabetes is defined as the positive event, and no diabetes is defined as the negative event. The confusion matrix for two classes is used to extract the values of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP indicates the tuples that correctly indicate diabetes. TN refers to the tuples that correctly indicate no diabetes. FP indicates the tuples that incorrectly indicate diabetes, and they are not diabetics. Finally, FN refers to the tuples that incorrectly indicate no diabetes, and they have diabetes. To measure the performance of the proposed model, we utilized the following metrics. Sensitivity is the proportion of true positives to all positive instances in the dataset; specificity is the proportion of true negatives to all negative instances. The classifier should be as sensitive and as specific as possible. Classification accuracy (CA) determines how well the classifier correctly identifies objects. Precision, or positive predictive value (PPV), is the proportion of cases with positive test results that are correctly classified. In addition, negative predictive value (NPV) is the proportion of cases with negative test results that are correctly classified. F-measure (FM) is the harmonic mean of precision and recall. The Matthews correlation coefficient (MCC) calculates the correlation between prediction and observation for the binary classification [70], as shown in Equation (20):

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \times (\mathrm{TP} + \mathrm{FN}) \times (\mathrm{TN} + \mathrm{FP}) \times (\mathrm{TN} + \mathrm{FN})}} \tag{20}$$

F-measure and MCC are critical because they measure the overall performance of a method. The false positive rate (FPR) is the inverse of specificity, indicating the proportion of negative instances that are erroneously classified as positive, as shown in Equation (21).

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{21}$$

The false negative rate (FNR) is the inverse of sensitivity, indicating the proportion of positive instances that are erroneously classified as negative, as shown in Equation (22):

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} \tag{22}$$

The error rate (ER), or misclassification rate, is the inverse of accuracy, giving the percentage of instances that are erroneously classified, as shown in Equation (23):

$$\mathrm{ER} = 1 - \mathrm{AC} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{23}$$

The geometric—means (GM) metric proposed by Kubat and Matwin [71] can also be used to evaluate classifiers as well, as shown in Equation (24). GM measures the balance between the classification performance of both the majority and the minority classes. A low GM indicates poor performance for the positive cases, even if the negative cases are correctly classified. GM avoids overfitting the negative class and under-fitting the positive class.

$$\mathrm{GM} = \sqrt{\mathrm{se} \times \mathrm{sp}} \tag{24}$$

5.2. Evaluation Results

In this section, we discuss the comparison between the proposed framework and other methods, including individual classifiers. We compared the ensemble model with other ensemble models in the literature, and with popular individual classifiers used in our combined model. Due to space restrictions, we used 10-fold cross-validation. The second issue is quantification. It determines what metrics will be used to measure classifier performance. This issue was discussed in the previous section. As mentioned earlier, we selected seven popular classifiers (NB, SVM, DT, FDT, ANN, LR, and KNN) from different domains to build a well-designed heterogeneous ensemble. To select the most effective algorithm with the most effective feature set for every sub-dataset, we evaluated all of the utilized base classifiers with every sub-dataset. We conducted this evaluation with the CFS and the wrapper FS techniques.

5.2.1. Base Classifier Evaluations Based on CFS

The base classifiers were executed with every sub-dataset by using the CFS technique. We constructed 63 different base classifiers (i.e., seven classifiers for nine sub-datasets). FDT was not applied to the categorical sub-datasets including symptoms, urine analysis, and diseases. Table 2 collects the performance metrics, including CA, Se, Sp, PPV, NPV, FM, MCC, FPR, FNR, ER, and GM. To make the comparison more straightforward, we compared the accuracy and F-measure of these algorithms for each sub-dataset. The other metrics were used to make more in-depth comparisons. As shown in Figures 4 and 5, we can select the best classifiers suitable for each specific sub-dataset. For demographics, DT had the best performance at 70% CA and 74.3% FM. For sugar lab tests, DT also had the best performance at 90% CA and 90.6% FM. For hematological profiles, LR had the best performance at 65% CA and 71.2% FM. For the symptoms sub-dataset, the classification performance of ANN outperformed other classifiers with 58.3% CA and 60.3% FM. For kidney function lab tests, DT had the best performance at 51.7% CA and 68.1% FM. The urine analysis sub-dataset saw better classification from the KNN algorithm, with 68.3% CA and 64.2% FM. For the lipid profiles, DT provided 63.3% CA and 73.2% FM as the most accurate. FDT had the best performance for liver function tests, with 61.7% CA and 51.1% FM. Finally, ANN had the best performance for the diseases sub-dataset, with 53.3% CA and 46.2% FM. All of these evaluations are based on the CFS technique and 10-fold cross-validation.

Table 2. The comparison of base classifiers for all datasets using CFS and 10-fold cross-validation.

Sub-Dataset	Algorithm	CA	Se	Sp	PPV	NPV	FM	MCC	FPR	FNR	ER	GM
Demographics	SVM	0.567	0.719	0.393	0.575	0.550	0.639	0.118	0.607	0.281	0.433	1.054
	DT	**0.700**	**0.813**	**0.571**	**0.684**	**0.727**	**0.743**	**0.397**	**0.429**	**0.188**	**0.300**	**1.177**
	NB	0.616	0.656	0.571	0.636	0.593	0.646	0.228	0.429	0.344	0.384	1.108
	ANN	0.650	0.781	0.500	0.641	0.667	0.704	0.294	0.500	0.219	0.350	1.132
	LR	0.650	0.719	0.571	0.657	0.640	0.687	0.294	0.429	0.281	0.350	1.136
	FDT	0.550	0.625	0.464	0.571	0.520	0.597	0.090	0.536	0.375	0.450	1.044
	KNN (k = 3)	0.500	0.563	0.429	0.529	0.462	0.545	0.009	0.571	0.438	0.500	0.996
Sugar lab tests	SVM	0.867	0.875	0.857	0.875	0.857	0.875	0.732	0.143	0.125	0.133	1.316
	DT	**0.900**	**0.906**	**0.893**	**0.906**	**0.893**	**0.906**	**0.799**	**0.107**	**0.094**	**0.100**	**1.341**
	NB	0.867	0.875	0.857	0.875	0.857	0.875	0.732	0.143	0.125	0.133	1.316
	ANN	0.867	0.844	0.893	0.900	0.833	0.871	0.735	0.107	0.156	0.133	1.318
	LR	0.833	0.813	0.857	0.867	0.800	0.839	0.668	0.143	0.188	0.167	1.292
	FDT	0.648	0.963	0.333	0.591	0.900	0.732	0.381	0.667	0.037	0.352	1.139
	KNN (k = 3)	0.833	0.813	0.857	0.867	0.800	0.839	0.668	0.143	0.188	0.167	1.292
Hematological profiles	SVM	0.633	0.781	0.464	0.625	0.650	0.694	0.260	0.536	0.219	0.367	1.116
	DT	0.617	0.750	0.464	0.615	0.619	0.676	0.224	0.536	0.250	0.383	1.102
	NB	0.650	0.750	0.536	0.649	0.652	0.696	0.293	0.464	0.250	0.350	1.134
	ANN	0.617	0.750	0.464	0.615	0.619	0.676	0.224	0.536	0.250	0.383	1.102
	LR	**0.650**	**0.813**	**0.464**	**0.634**	**0.684**	**0.712**	**0.297**	**0.536**	**0.188**	**0.350**	**1.130**
	FDT	0.383	0.563	0.464	0.439	0.433	0.493	0.270	0.821	0.531	0.617	1.014
	KNN (k = 3)	0.617	0.750	0.464	0.615	0.619	0.676	0.224	0.536	0.250	0.383	1.102
Symptoms	SVM	0.550	0.656	0.429	0.568	0.522	0.609	0.087	0.571	0.344	0.450	1.041
	DT	0.467	0.531	0.393	0.500	0.423	0.515	0.076	0.607	0.469	0.533	0.961
	NB	0.567	0.594	0.536	0.594	0.536	0.594	0.129	0.464	0.406	0.433	1.063
	ANN	**0.583**	**0.594**	**0.571**	**0.613**	**0.552**	**0.603**	**0.165**	**0.429**	**0.406**	**0.417**	**1.080**
	LR	0.567	0.625	0.500	0.588	0.538	0.606	0.126	0.500	0.375	0.433	1.061
	KNN (k = 3)	0.500	0.344	0.679	0.550	0.475	0.423	0.024	0.321	0.656	0.500	1.011
Kidney function lab tests	SVM	0.467	0.875	0.000	0.500	0.000	0.636	0.250	1.000	0.125	0.533	0.935
	DT	**0.517**	**0.969**	**0.000**	**0.525**	**0.000**	**0.681**	**0.122**	**1.000**	**0.031**	**0.483**	**0.984**
	NB	0.417	0.688	0.107	0.468	0.231	0.557	0.249	0.893	0.313	0.583	0.892
	ANN	0.417	0.625	0.179	0.465	0.294	0.533	0.217	0.821	0.375	0.583	0.896
	LR	0.450	0.500	0.393	0.485	0.407	0.492	0.107	0.607	0.500	0.550	0.945
	FDT	0.467	0.500	0.429	0.500	0.429	0.500	0.071	0.571	0.500	0.533	0.964
	KNN (k = 3)	0.467	0.375	0.571	0.500	0.444	0.429	0.055	0.429	0.625	0.533	0.973
Lipid profiles	SVM	0.500	0.875	0.071	0.519	0.333	0.651	0.089	0.929	0.125	0.500	0.973
	DT	**0.633**	**0.938**	**0.286**	**0.600**	**0.800**	**0.732**	**0.299**	**0.714**	**0.063**	**0.367**	**1.106**
	NB	0.517	0.281	0.786	0.600	0.489	0.383	0.077	0.214	0.719	0.483	1.033
	ANN	0.567	0.844	0.250	0.563	0.583	0.675	0.117	0.750	0.156	0.433	1.046
	LR	0.517	0.719	0.286	0.535	0.471	0.613	0.005	0.714	0.281	0.483	1.002
	FDT	0.467	0.438	0.250	0.500	0.333	0.467	0.063	0.500	0.438	0.533	0.829
	KNN (k = 3)	0.567	0.875	0.214	0.560	0.600	0.683	0.120	0.786	0.125	0.433	1.044
Urine analysis	SVM	0.667	0.500	0.857	0.800	0.600	0.615	0.378	0.143	0.500	0.333	1.165
	DT	0.683	0.531	0.857	0.810	0.615	0.642	0.406	0.143	0.469	0.317	1.178
	NB	0.667	0.500	0.857	0.800	0.600	0.615	0.378	0.143	0.500	0.333	1.165
	ANN	0.700	0.500	0.929	0.889	0.619	0.640	0.467	0.071	0.500	0.300	1.195
	LR	0.667	0.469	0.893	0.833	0.595	0.600	0.394	0.107	0.531	0.333	1.167
	KNN (k = 3)	**0.683**	**0.531**	**0.857**	**0.810**	**0.615**	**0.642**	**0.406**	**0.143**	**0.469**	**0.317**	**1.178**
Liver function tests	SVM	0.483	0.813	0.107	0.510	0.333	0.627	0.112	0.893	0.188	0.517	0.959
	DT	0.417	0.594	0.214	0.463	0.316	0.521	0.206	0.786	0.406	0.583	0.899
	NB	0.533	0.250	0.857	0.667	0.500	0.364	0.134	0.143	0.750	0.467	1.052
	ANN	0.500	0.531	0.464	0.531	0.464	0.531	0.004	0.536	0.469	0.500	0.998
	LR	0.550	0.469	0.643	0.600	0.514	0.526	0.113	0.357	0.531	0.450	1.054
	FDT	**0.617**	**0.375**	**0.405**	**0.800**	**0.556**	**0.511**	**0.309**	**0.107**	**0.625**	**0.383**	**0.883**
	KNN (k = 3)	0.583	0.500	0.679	0.640	0.543	0.561	0.181	0.321	0.500	0.417	1.086
Diseases	SVM	0.517	0.438	0.607	0.560	0.486	0.491	0.045	0.393	0.563	0.483	1.022
	DT	0.450	0.375	0.536	0.480	0.429	0.421	0.090	0.464	0.625	0.550	0.954
	NB	0.600	0.500	0.714	0.667	0.556	0.571	0.218	0.286	0.500	0.400	1.102
	ANN	**0.533**	**0.375**	**0.714**	**0.600**	**0.500**	**0.462**	**0.094**	**0.286**	**0.625**	**0.467**	**1.044**
	LR	0.483	0.281	0.714	0.529	0.465	0.367	0.005	0.286	0.719	0.517	0.998
	KNN	0.466	0.344	0.607	0.500	0.447	0.407	0.051	0.393	0.656	0.534	0.975

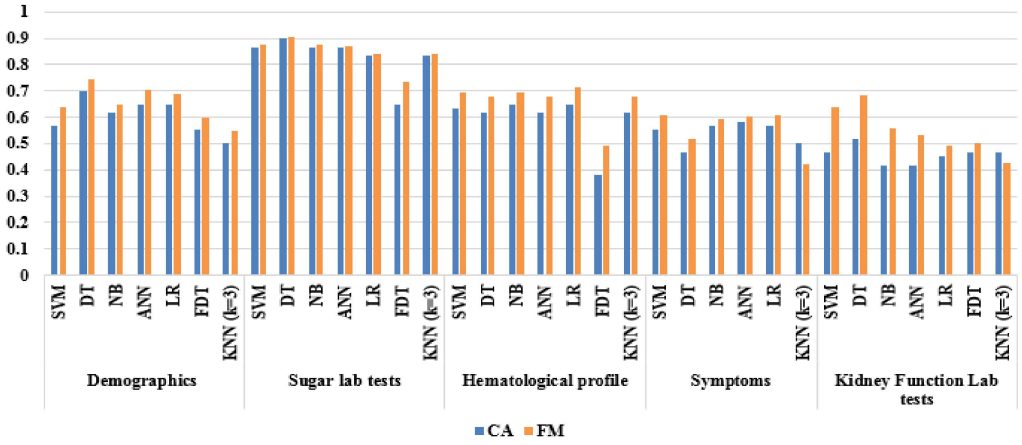


Figure 4. A comparison between CA and FM for base classifiers using CFS (part 1).

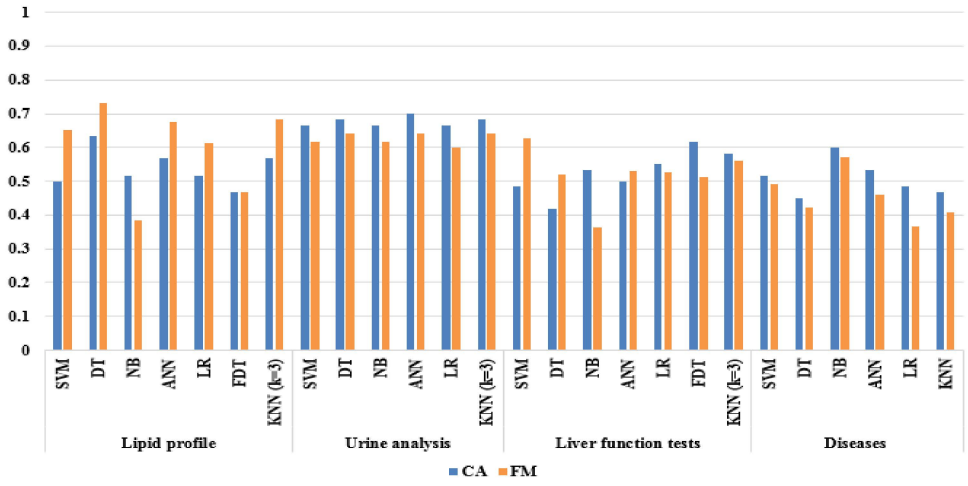


Figure 5. A comparison between CA and FM for base classifiers using CFS (part 2).

5.2.2. Base Classifier Evaluation Based on Wrapper FS

In this section, we evaluate the set of base classifiers on every sub-dataset and register the results. The wrapper FS algorithm was applied first to determine the most suitable feature subset, and the selected features were then used to train and test every classifier. We constructed 63 different base classifiers (i.e., seven classifiers for nine sub-datasets). FDT was not applied to the categorical sub-datasets including symptoms, urine analysis, and diseases. Table 3 collects all relevant performance metrics, including CA, FM, Se, Sp, MCC, etc., for each algorithm on all datasets. As before, we collected the best base classifiers for all sub-datasets. We concentrated on CA and FM for the comparison between different algorithms.

As shown in Figures 6 and 7, we can select the best classifiers suitable for each specific sub-dataset. For demographics, SVM had the best performance at 70% CA and 74.4% FM. For sugar lab tests, DT had the best performance at 90% CA and 90.6% FM. For the hematological profiles, NB had the best performance with 66.7% CA and 70.6% FM. For the symptoms sub-dataset, the classification performance of SVM outperformed other classifiers, with 61.7% CA and 56.6% FM. For the kidney function lab tests, DT had the best performance at 53.3% CA and 69.6% FM. The urine analysis sub-dataset obtained better classification with the LR algorithm, at 73.3% CA and 66.7% FM. For the lipid profiles, ANN provided 66.7% CA and 74.4% FM, which were the most accurate.

Table 3. The comparison of base classifiers for all datasets using wrapper FS and 10-fold cross-validation.

Dataset	Algorithm	CA	Se	Sp	PPV	NPV	FM	MCC	FPR	FNR	ER	GM
Demographics	SVM	**0.700**	**0.813**	**0.571**	**0.684**	**0.727**	**0.743**	**0.397**	**0.429**	**0.188**	**0.300**	**1.177**
	DT	0.667	0.906	0.393	0.630	0.786	0.744	0.353	0.607	0.094	0.333	1.140
	NB	0.650	0.688	0.607	0.667	0.630	0.677	0.295	0.393	0.313	0.350	1.138
	ANN	0.700	0.813	0.571	0.684	0.727	0.743	0.397	0.429	0.188	0.300	1.177
	LR	0.633	0.688	0.571	0.647	0.615	0.667	0.261	0.429	0.313	0.367	1.122
	FDT	0.483	0.625	0.321	0.513	0.429	0.563	0.056	0.679	0.375	0.517	0.973
	KNN (k = 3)	0.650	0.656	0.643	0.677	0.621	0.667	0.299	0.357	0.344	0.350	1.140
Sugar lab tests	SVM	0.867	0.875	0.857	0.875	0.857	0.875	0.732	0.143	0.125	0.133	1.316
	DT	**0.900**	**0.906**	**0.893**	**0.906**	**0.893**	**0.906**	**0.799**	**0.107**	**0.094**	**0.100**	**1.341**
	NB	0.883	0.875	0.893	0.903	0.862	0.889	0.767	0.107	0.125	0.117	1.330
	ANN	0.883	0.875	0.643	0.903	0.621	0.889	0.767	0.107	0.344	0.117	1.232
	LR	0.850	0.844	0.857	0.871	0.828	0.857	0.700	0.143	0.156	0.150	1.304
	FDT	0.648	0.963	0.333	0.591	0.900	0.732	0.381	0.667	0.037	0.352	1.139
	KNN (k = 3)	0.867	0.844	0.893	0.900	0.833	0.871	0.735	0.107	0.156	0.133	1.318
Hematological profiles	SVM	0.600	0.781	0.393	0.595	0.611	0.676	0.190	0.607	0.219	0.400	1.083
	DT	0.567	0.719	0.393	0.575	0.550	0.639	0.118	0.607	0.281	0.433	1.054
	NB	**0.667**	**0.750**	**0.571**	**0.667**	**0.667**	**0.706**	**0.327**	**0.429**	**0.250**	**0.333**	**1.150**
	ANN	0.650	0.750	0.536	0.649	0.652	0.696	0.293	0.464	0.250	0.350	1.134
	LR	0.633	0.813	0.429	0.619	0.667	0.703	0.262	0.571	0.188	0.357	1.114
	FDT	0.417	0.594	0.214	0.463	0.316	0.521	0.206	0.786	0.406	0.583	0.899
	KNN (k = 3)	0.583	0.625	0.536	0.606	0.556	0.615	0.161	0.464	0.375	0.417	1.077
Symptoms	SVM	**0.617**	**0.469**	**0.786**	**0.714**	**0.564**	**0.566**	**0.266**	**0.214**	**0.531**	**0.383**	**1.120**
	DT	0.467	0.375	0.571	0.500	0.444	0.429	0.055	0.429	0.625	0.533	0.973
	NB	0.550	0.563	0.536	0.581	0.517	0.571	0.098	0.464	0.438	0.450	1.048
	ANN	0.467	0.375	0.571	0.500	0.444	0.429	0.055	0.429	0.625	0.533	0.973
	LR	0.550	0.625	0.464	0.571	0.520	0.597	0.090	0.536	0.375	0.450	1.044
	KNN (k = 3)	0.500	0.563	0.429	0.529	0.462	0.545	0.009	0.571	0.438	0.500	0.996
Kidney function lab tests	SVM	0.500	0.875	0.071	0.519	0.333	0.651	0.089	0.929	0.125	0.500	0.973
	DT	**0.533**	**1.000**	**0.000**	**0.533**	**0.000**	**0.696**	**0.000**	**1.000**	**0.000**	**0.467**	**1.000**
	NB	0.433	0.531	0.321	0.472	0.375	0.500	0.150	0.679	0.469	0.567	0.923
	ANN	0.450	0.625	0.447	0.488	0.586	0.548	0.134	0.750	0.375	0.550	1.036
	LR	0.417	0.531	0.286	0.459	0.348	0.493	0.188	0.714	0.469	0.583	0.904
	FDT	0.467	0.594	0.321	0.500	0.409	0.543	0.088	0.679	0.406	0.533	0.957
	KNN (k = 3)	0.533	0.469	0.607	0.577	0.500	0.517	0.076	0.393	0.531	0.467	1.037
Lipid profiles	SVM	0.533	1.000	0.000	0.533	DIV/0!	0.696	0.000	1.000	0.000	0.457	1.000
	DT	0.600	0.938	0.214	0.577	0.750	0.714	0.223	0.786	0.063	0.400	1.073
	NB	0.517	0.375	0.321	0.571	0.310	0.453	0.056	0.321	0.625	0.483	0.835
	ANN	**0.667**	**0.906**	**0.393**	**0.630**	**0.786**	**0.744**	**0.353**	**0.607**	**0.094**	**0.333**	**1.140**
	LR	0.617	0.719	0.500	0.622	0.609	0.667	0.224	0.500	0.281	0.383	1.104
	FDT	0.483	0.531	0.429	0.515	0.444	0.523	0.040	0.571	0.469	0.517	0.980
	KNN (k = 3)	0.600	0.594	0.607	0.633	0.567	0.613	0.200	0.393	0.406	0.400	1.096
Urine analysis	SVM	0.717	0.469	1.000	1.000	0.622	0.638	0.540	0.000	0.531	0.283	1.212
	DT	0.683	0.531	0.857	0.810	0.615	0.642	0.406	0.143	0.469	0.317	1.178
	NB	0.650	0.500	0.821	0.762	0.590	0.604	0.336	0.179	0.500	0.350	1.150
	ANN	0.717	0.500	0.964	0.941	0.628	0.653	0.514	0.036	0.500	0.283	1.210
	LR	**0.733**	**0.500**	**1.000**	**1.000**	**0.636**	**0.667**	**0.564**	**0.000**	**0.500**	**0.267**	**1.225**
	KNN (k = 3)	0.683	0.438	0.964	0.933	0.600	0.596	0.463	0.036	0.563	0.317	1.184
Liver function tests	SVM	0.417	0.625	0.179	0.465	0.294	0.533	0.217	0.821	0.375	0.583	0.896
	DT	0.450	0.656	0.214	0.488	0.353	0.560	0.143	0.786	0.344	0.550	0.933
	NB	0.483	0.250	0.750	0.533	0.467	0.340	0.000	0.250	0.750	0.517	1.000
	ANN	0.617	0.500	0.391	0.696	0.568	0.582	0.257	0.250	0.500	0.383	0.944
	LR	0.500	0.625	0.357	0.526	0.455	0.571	0.018	0.643	0.375	0.500	0.991
	FDT	**0.650**	**0.438**	**0.893**	**0.824**	**0.581**	**0.571**	**0.366**	**0.107**	**0.563**	**0.350**	**1.154**
	KNN (k = 3)	0.617	0.688	0.536	0.629	0.600	0.657	0.226	0.464	0.313	0.383	1.106
Diseases	SVM	**0.657**	**0.313**	**0.857**	**0.714**	**0.522**	**0.435**	**0.200**	**0.143**	**0.688**	**0.433**	**1.082**
	DT	0.483	0.375	0.607	0.522	0.459	0.436	0.018	0.393	0.625	0.517	0.991
	NB	0.600	0.500	0.714	0.667	0.556	0.571	0.218	0.286	0.500	0.400	1.102
	ANN	0.567	0.406	0.750	0.650	0.525	0.500	0.165	0.250	0.594	0.433	1.075
	LR	0.533	0.406	0.679	0.591	0.500	0.481	0.088	0.321	0.594	0.467	1.041
	KNN (k = 3)	0.517	0.469	0.571	0.556	0.485	0.508	0.040	0.429	0.531	0.483	1.020

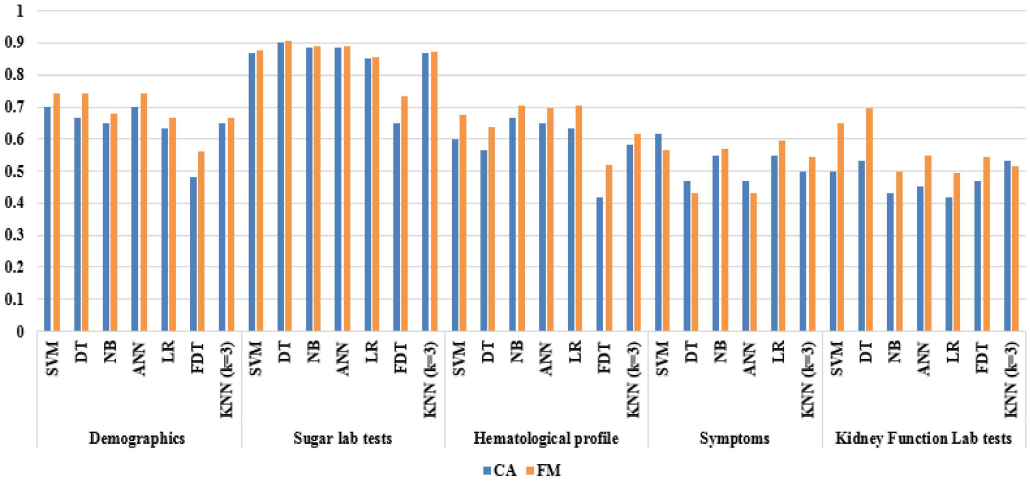


Figure 6. A comparison between CA and FM for base classifiers using wrapper FS (part 1).

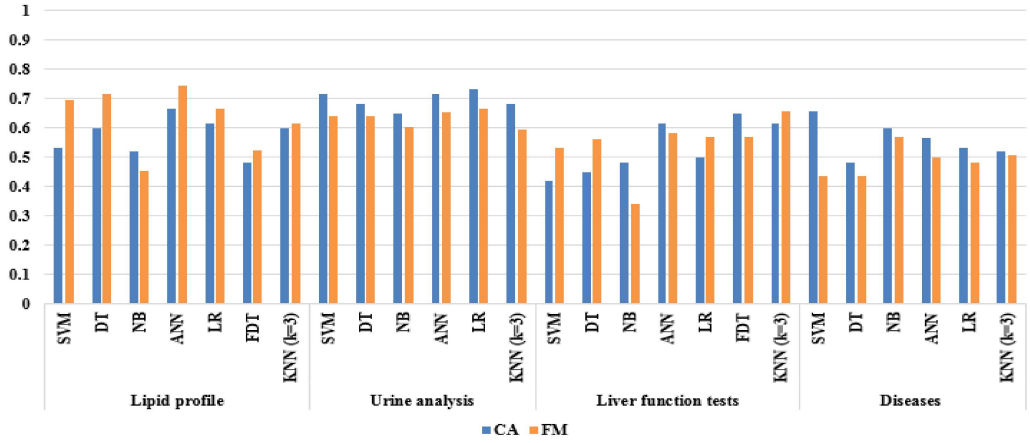


Figure 7. A comparison between CA and FM for base classifiers using wrapper FS (part 2).

FDT had the best performance for the liver function tests, with 65% CA and 57.1% FM. Finally, SVM had the best performance for the diseases sub-dataset, at 65.7% CA and 43.5% FM. All of these evaluations are based on the wrapper FS technique and 10-fold cross-validation.

From the previous comprehensive evaluations in Tables 2 and 3, we determined the optimum base classifier and the most suitable features for every sub-dataset. FM has a higher priority than other metrics because it is used as the weight of each base classifier. Table 4 lists the utilized base algorithms, their selected features, and their weights for the nine datasets.

Table 4. The proposed ensemble classifier's base algorithms and their weights.

No.	Dataset	Base Algorithm	FS Technique	Weight (FM)
1	Demographics	SVM	Wrapper	74.4
2	Sugar lab tests	DT	Correlation FS	90.6
3	Hematological profiles	LR	Correlation FS	71.2
4	Symptoms	ANN	Correlation FS	60.3
5	Kidney function Lab tests	DT	Wrapper	69.6
6	Lipid profile	ANN	Wrapper	66.7
7	Urine analysis	LR	Wrapper	74.4
8	Liver function tests	FDT	Wrapper	57.1
9	Diseases	ANN	Correlation FS	46.2

5.2.3. The Proposed Ensemble Evaluation

To evaluate the proposed algorithm, we utilized WEKA's JAVA APIs to customize the implementation process according to the results in Table 4. The proposed framework achieved the best overall performance for overall base classifiers. The framework has a recall of 0.902, CA of 0.900, specificity of 0.895, precision of 0.949, NPV of 0.810, FM of 0.925, FPR of 0.105, FNR of 0.098, ER of 0.100, MCC of 0.778, and GM of 1.341.

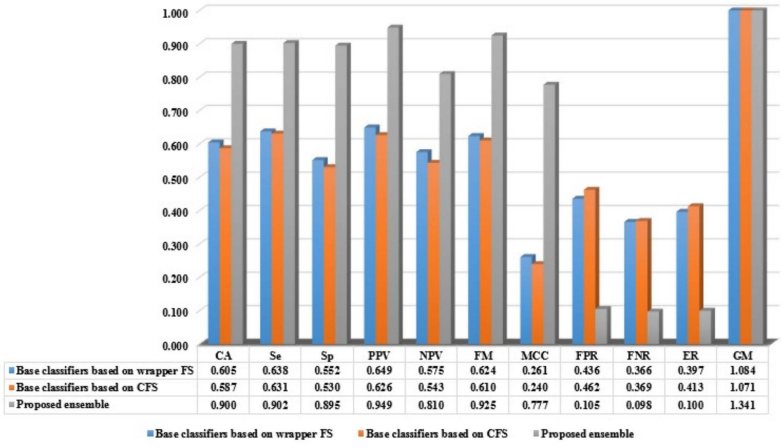
These results are very logical because when we decide who has diabetes, we take all of the patient's profile into consideration. For example, we can see that the level of glucose in the blood can provide accurate results in the diagnosis process; but medically, there are many reasons other than diabetes for an increase in glucose level in the blood.

As a result, taking a decision based on the level of glucose only seems to provide inaccurate results. At the time of diagnosis, patients with diabetes often have complications so that these complications can add value to the diagnosis process. This is exactly what we do in this framework.

The patient's symptoms, demographics, diseases, liver tests, kidney tests, lipid profile, and urine analysis are considered in the diagnosis process.

The proposed ensemble classifier achieves this performance as a result of several steps: (i) the dataset is completely preprocessed; (ii) the whole dataset is medically divided into correlated features; (iii) the most suitable base classifier is selected for each sub-dataset; (iv) the best feature vector is selected for each base classifier in an accurate way; and (v) the base classifiers are weighted based on FM, which is the harmonic mean of precision and recall.

Performance of the proposed ensemble was compared with the average performance of single classifiers in Tables 2 and 3. Figure 8 illustrates that our framework outperforms all of the base classifiers, including the CFS-based and wrapper-based algorithms. Regarding the computational complexity of the proposed classifier, its complexity is $O_{proposed} = max(O_{SVM} + O_{KNN} + O_{NB} + O_{DT} + O_{FDT} + O_{ANN} + O_{LR})$ because it runs the base algorithms in parallel. Because m < n, O_{SVM} is the largest complexity. As a result, the $O_{proposed}$ is equal to n^3.



	CA	Se	Sp	PPV	NPV	FM	MCC	FPR	FNR	ER	GM
Base classifiers based on wrapper FS	0.605	0.638	0.552	0.649	0.575	0.624	0.261	0.436	0.366	0.397	1.084
Base classifiers based on CFS	0.587	0.631	0.530	0.626	0.543	0.610	0.240	0.462	0.369	0.413	1.071
Proposed ensemble	0.900	0.902	0.895	0.949	0.810	0.925	0.777	0.105	0.098	0.100	1.341

Figure 8. Comparison between the proposed framework and average results of base classifiers.

To compare the proposed ensemble with the other ensembles, we evaluated a set of meta-classifiers, including homogeneous ensembles (i.e., bagging, boosting, and RF) and heterogeneous ensembles (i.e., voting and stacking) for every sub-dataset by using CFS. We created 45 meta-classifiers (i.e., five classifiers for nine datasets). These simple ensembles failed to improve overall performance. For example, in the demographics dataset, the base classifier SVM in Table 3 achieves performance similar to all ensemble algorithms for the same dataset in Table 5. For each sub-dataset, we used the most suitable setting for the meta-classifier. For example for the demographic dataset, we use DT for the bagging technique; four classifiers (LR, SVM, NB, and DT) used majority voting for

the voting technique; and AdaboostM1 utilized LR. These settings achieved the best performance for meta-classifiers.

Table 5. Classification results for ensemble classifiers with correlation-based feature selection.

Sub-Dataset	Algorithm	CA	Se	Sp	PPV	NPV	FM	MCC	FPR	FNR	ER	GM
Demographics	RF	0.617	0.719	0.500	0.622	0.609	0.667	0.224	0.500	0.281	0.383	1.104
	Bagging	0.667	0.781	0.536	0.658	0.682	0.714	0.328	0.464	0.219	0.333	1.147
	Voting	0.650	0.688	0.607	0.667	0.630	0.677	0.295	0.393	0.313	0.530	1.138
	Stacking	0.567	0.500	0.643	0.615	0.529	0.552	0.144	0.357	0.500	0.433	1.069
	AdaBoostM1	0.700	0.781	0.607	0.694	0.708	0.735	0.396	0.393	0.219	0.300	1.178
Sugar lab tests	RF	0.867	0.844	0.893	0.900	0.833	0.871	0.735	0.107	0.156	0.133	1.318
	Bagging	0.867	0.875	0.857	0.875	0.857	0.875	0.732	0.143	0.125	0.133	1.316
	Voting	0.867	0.875	0.857	0.875	0.857	0.875	0.732	0.143	0.125	0.133	1.316
	Stacking	0.850	0.844	0.857	0.871	0.828	0.857	0.700	0.143	0.156	0.150	1.304
	AdaBoostM1	0.867	0.844	0.893	0.900	0.833	0.871	0.735	0.107	0.156	0.133	1.318
Hematological profiles	RF	0.617	0.750	0.464	0.615	0.619	0.676	0.224	0.536	0.250	0.383	1.102
	Bagging	0.633	0.750	0.500	0.632	0.636	0.686	0.259	0.500	0.250	0.367	1.118
	Voting	0.617	0.750	0.464	0.615	0.619	0.676	0.224	0.536	0.250	0.383	1.102
	Stacking	0.533	0.563	0.500	0.563	0.500	0.563	0.063	0.500	0.438	0.467	1.031
	AdaBoostM1	0.650	0.813	0.464	0.634	0.684	0.712	0.297	0.536	0.188	0.350	1.130
Symptoms	RF	0.567	0.531	0.607	0.607	0.531	0.567	0.138	0.393	0.469	0.433	1.067
	Bagging	0.500	0.563	0.429	0.529	0.462	0.545	0.009	0.571	0.438	0.500	0.996
	Voting	0.517	0.531	0.500	0.548	0.483	0.540	0.031	0.500	0.469	0.483	1.015
	Stacking	0.467	0.563	0.357	0.500	0.417	0.529	0.082	0.643	0.438	0.533	0.959
	AdaBoostM1	0.533	0.594	0.464	0.559	0.500	0.576	0.058	0.536	0.406	0.467	1.029
Kidney function lab tests	RF	0.400	0.344	0.464	0.423	0.382	0.379	0.193	0.536	0.656	0.600	0.899
	Bagging	0.417	0.563	0.250	0.462	0.333	0.507	0.196	0.750	0.438	0.583	0.902
	Voting	0.483	0.906	0.000	0.509	0.000	0.652	0.215	1.000	0.094	0.517	0.952
	Stacking	0.550	0.750	0.321	0.558	0.529	0.640	0.079	0.679	0.250	0.450	1.035
	AdaBoostM1	0.517	0.969	0.000	0.525	0.000	0.681	0.122	1.000	0.031	0.483	0.984
Lipid profiles	RF	0.650	0.906	0.357	0.617	0.769	0.734	0.319	0.643	0.094	0.350	1.124
	Bagging	0.533	0.750	0.286	0.545	0.500	0.632	0.040	0.714	0.250	0.467	1.018
	Voting	0.600	0.875	0.286	0.583	0.667	0.700	0.200	0.714	0.125	0.400	1.077
	Stacking	0.583	0.781	0.357	0.581	0.588	0.667	0.153	0.643	0.219	0.417	1.067
	AdaBoostM1	0.633	0.969	0.250	0.596	0.875	0.738	0.321	0.750	0.031	0.367	1.104
Urine analysis	RF	0.667	0.500	0.857	0.800	0.600	0.615	0.378	0.143	0.500	0.333	1.165
	Bagging	0.667	0.500	0.857	0.800	0.600	0.615	0.378	0.143	0.500	0.333	1.165
	Voting	0.683	0.531	0.857	0.810	0.615	0.642	0.406	0.143	0.469	0.317	1.178
	Stacking	0.583	0.406	0.786	0.684	0.537	0.510	0.206	0.214	0.594	0.417	1.092
	AdaBoostM1	0.667	0.500	0.857	0.800	0.600	0.615	0.378	0.143	0.500	0.333	1.165
Liver function tests	RF	0.500	0.500	0.500	0.533	0.467	0.516	0.000	0.500	0.500	0.500	1.000
	Bagging	0.533	0.250	0.857	0.667	0.500	0.364	0.134	0.143	0.750	0.467	1.052
	Voting	0.550	0.438	0.679	0.609	0.514	0.509	0.119	0.321	0.563	0.450	1.057
	Stacking	0.517	0.531	0.500	0.548	0.483	0.540	0.031	0.500	0.469	0.483	1.015
	AdaBoostM1	0.533	0.250	0.857	0.667	0.500	0.364	0.134	0.143	0.750	0.467	1.052
Diseases	RF	0.500	0.406	0.607	0.542	0.472	0.464	0.014	0.393	0.594	0.500	1.007
	Bagging	0.533	0.531	0.536	0.567	0.500	0.548	0.067	0.464	0.469	0.467	1.033
	Voting	0.550	0.406	0.714	0.619	0.513	0.491	0.126	0.286	0.594	0.450	1.058
	Stacking	0.483	0.500	0.464	0.516	0.448	0.508	0.036	0.536	0.500	0.517	0.982
	AdaBoostM1	0.600	0.500	0.714	0.667	0.556	0.571	0.218	0.286	0.500	0.400	1.102

We evaluated the above ensemble classifiers based on the wrapper FS technique; however, they provided results somewhat comparable to the CFS technique. Figure 9 shows a comparison between the proposed classifier and the maximum values of the five ensembles in Table 5. As we can see, the proposed ensemble achieves overall improved performance and low error rates. At the same time, these results are medically acceptable and get high confidence from physicians, because all of the patient's characteristics are included in the decision-making process. As a result, the proposed method can be applied in similar problems to provide classifiers of other diseases. We work very closely with two medical experts to prepare and implement this study. The domain experts validated the collected

datasets, guided in data preprocessing and understanding the disease intuition, and tested the final system. In addition, the results of the system have been validated by domain experts.

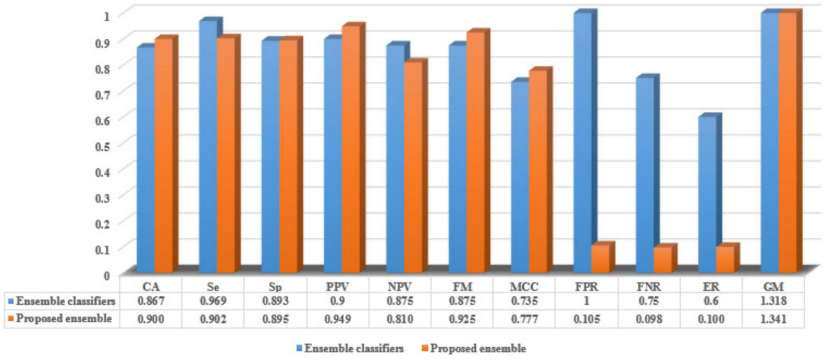


Figure 9. A comparison between the proposed ensemble and maximum results of other ensembles.

Although the proposed model achieves promising results, it has some limitations that will be handed in future work. For example, the model has not been tested on other datasets. The available public diabetes datasets (e.g. PIDD) are not multimodal data. They have not clearly separated groups of features to be used as complementary multimodal data in the proposed model. Further, the proposed model has not handled the semantic relations between medical concepts such as diseases and symptoms. This issue can be handled using semantic data mining techniques by embedding ontology reasoning in the learning process. In addition, as diabetes is a chronic disease, it is normal to find many readings for each feature in different time. These data could be collected from sensors connected to the patient body [72]. These temporal data need special analysis, which can benefit in remote patient monitoring.

6. Conclusions

This paper proposed a heterogeneous ensemble classifier to improve disease detection accuracy. The proposed classifier was applied to a serious chronic disease: DM. To take best advantage of single classifiers for designing the proposed classifier and to produce better results than any of the single classifiers, we first selected a set of diverse, well-known, and heavily applied algorithms in the medical field: SVM, FDT, ANN, NB, LR, DT, and KNN. Second, we used two well-known feature selection techniques (CFS and wrapper FS) to select the most suitable features for every algorithm with every sub-dataset. Third, we trained all algorithms with all the preprocessed sub-datasets. Finally, we built the proposed algorithm using the base classifiers with the best results. The proposed ensemble was evaluated and tested. It achieved a recall of 90.2%, CA of 90%, specificity of 89.5%, precision of 94.9%, NPV of 81%, FM of 92.5%, FPR of 10.5%, FNR of 9.8%, ER of 10%, MCC of 77.8%, and GM of 1.341.

These results outperformed the average performance of base classifiers and other ensembles. This study has demonstrated that a well-designed heterogeneous ensemble classifier can be more accurate than any other classifier in disease detection; herein lies the main contribution of this study. In future work, we will extend the proposed ensemble to handle the semantic aspects of medical data. There is a possibility of using an OWL ontology and description logic semantics to achieve this goal. In addition, because diabetes is a chronic disease, it is critical to handle time dimensions in the patient data. Based on the promising results of the proposed framework, we will check it with other datasets and for diagnosis of other diseases. Analyzing the clinical "omics" data is very critical in clinical domain especially for disease treatment (http://omics.org/). In the future, we will study the relationship between diabetes and taken drugs based on the integration of regular medical data and genomic data.

Author Contributions: All authors participated equally in the design and implementation processes of this manuscript. They all participated in drafting the article or revising it critically for important intellectual content. Final approval of the version to be submitted was given by all authors. Author S.E.-S. provided the formal

analysis of diabetes diagnoses medical problem, collected the medical dataset, and perform the necessary data preprocessing. Authors M.E. and F.A. provided the conceptualization of the model by designing the proposed framework. They performed the feature selection steps. Authors T.A. and S.M.R.I. write the proposed algorithm and implement the software of the proposed system in Java. Author K.-S.K. reviewed the proposed model and its implementation; he provided the validation of the model and collected the final results of the system. All authors read and approved the final manuscript.

Funding: This work was supported by National Research Foundation of Korea-Grant funded by the Korean Government (Ministry of Science and ICT)-NRF-2017R1A2B2012337).

Acknowledgments: The authors would also like to thank Farid Badria, a professor of pharmacognosy and head of the Liver Research Lab, Mansoura University, Egypt, and Hosam Zaghloul, a professor in the Clinical Pathology Department, Faculty of Medicine, Mansoura University, Egypt, for their efforts to assist this work.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Zarkogianni, K.; Litsa, E.; Mitsis, K.; Wu, P.Y.; Kaddi, C.D.; Cheng, C.W.; Wang, M.D.; Nikita, K.S. A Review of Emerging Technologies for the Management of Diabetes Mellitus. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 2735–2749. [CrossRef] [PubMed]
2. Upadhyaya, S.; Farahmand, K.; Baker-Demaray, T. Comparison of NN and LR classifiers in the context of screening native American elders with diabetes. *Expert Syst. Appl.* **2013**, *40*, 5830–5838. [CrossRef]
3. Guariguata, L.; Whiting, D.; Hambleton, I.; Beagley, J.; Linnenkamp, U.; Shaw, J. Global estimates of diabetes prevalence in adults for 2013 and projections for 2035 for the IDF Diabetes Atlas. *Diabetes Res. Clin. Pract.* **2014**, *2*, 137–149. [CrossRef] [PubMed]
4. Zheng, T.; Xie, W.; Xu, L.; He, X.; Zhang, Y.; You, M.; Yang, G.; Chen, Y. A machine learning-based framework to identify type 2 diabetes through electronic health records. *Int. J. Med. Inform.* **2017**, *97*, 120–127. [CrossRef] [PubMed]
5. Tripathi, B.; Srivastava, A. Diabetes mellitus complications and therapeutics. *Med. Sci Monit.* **2006**, *12*, RA130–RA147. [PubMed]
6. Heydari, M.; Teimouri, M.; Heshmati, Z.; Alavinia, S. Comparison of various classification algorithms in the diagnosis of type 2 diabetes in Iran. *Int. J. Diabetes Dev. Ctries.* **2016**, *36*, 167–173. [CrossRef]
7. Wei, W.Q.; Leibson, C.L.; Ransom, J.E.; Kho, A.N.; Chute, C.G. The absence of longitudinal data limits the accuracy of high-throughput clinical phenotyping for identifying type 2 diabetes mellitus subjects. *Int. J. Med. Inf.* **2013**, *82*, 239–247. [CrossRef]
8. Bashir, S.; Qamar, U.; Khan, F. IntelliHealth: A medical decision support application using a novel weighted multi-layer classifier ensemble framework. *J. Biomed. Inf.* **2016**, *59*, 185–200. [CrossRef]
9. Kavakiotis, I.; Tsave, O.; Salifoglou, A.; Maglaveras, N.; Vlahavas, I.; Chouvarda, I. Machine Learning and Data Mining Methods in Diabetes Research. *Comput. Struct. Biotechnol. J.* **2017**, *15*, 104–116. [CrossRef]
10. Meng, X.; Huang, Y.; Rao, D.; Zhang, Q.; Liu, Q. Comparison of three data mining models for predicting diabetes or prediabetes by risk factors. *Kaohsiung J. Med. Sci.* **2013**, *29*, 93–99. [CrossRef]
11. Marinov, M.; Mosa, A.; Yoo, I.; Boren, S.A. Data mining technologies for diabetes: A systematic review. *J. Diabetes Sci. Technol.* **2011**, *5*, 1549–1556. [CrossRef] [PubMed]
12. Mani, S.; Chen, Y.; Elasy, T.; Clayton, W.; Denny, J. Type 2 diabetes risk forecasting from EMR data using machine learning. In *AMIA Annual Symposium Proceeding*; American Medical Informatics Association: Bethesda, MD, USA, 2012; p. 606.
13. Zhu, J.; Xie, Q.; Zheng, K. An improved early detection method of type-2 diabetes mellitus using multiple classifier system. *Inf. Sci.* **2015**, *292*, 1–14. [CrossRef]
14. Huang, G.; Huang, K.; Lee, T.; Weng, J. An interpretable rule-based diagnostic classification of diabetic nephropathy among type 2 diabetes patients. *BMC Bioinform.* **2015**, *16* (Suppl. 1), S5. [CrossRef]
15. Noble, D.; Mathur, R.; Dent, T.; Meads, C.; Greenhalgh, T. Risk models and scores for type 2 diabetes: Systematic review. *BMJ* **2011**, *343*, d7163. [CrossRef] [PubMed]
16. American Diabetes Association. Screening for type 2 diabetes. *Diabetes Care* **2004**, *27* (Suppl. 1), s11–s14. [CrossRef] [PubMed]
17. Parvin, H.; MirnabiBaboli, M.; Alinejad-Rokny, H. Proposing a classifier ensemble framework based on classifier selection and decision tree. *Eng. Appl. Artif. Intell.* **2015**, *37*, 34–42. [CrossRef]

18. Sluban, B.; Lavrac, N. Relating ensemble diversity and performance: A study in class noise detection. *Neurocomputing* **2015**, *160*, 120–131. [CrossRef]
19. Kuncheva, L. *Combining Pattern Classifiers: Methods and Algorithm*, 2nd ed.; Wiley: New York, NY, USA, 2014.
20. Dietterich, T. Ensemble methods in machine learning. In Proceedings of the 1st International workshop on Multiple Classifier Systems (MCS 2000), Cagliary, Italy, 21–23 June 2000; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1857, pp. 1–15.
21. Patil, M.; Joshi, R.; Toshniwal, D. Hybrid prediction model for Type-2 diabetic patients. *Expert Syst. Appl.* **2010**, *37*, 8102–8108. [CrossRef]
22. Sanakal, R.; Jayakumari, S. Prognosis of Diabetes Using Data mining Approach-Fuzzy C Means Clustering and Support Vector Machine. *Int. J. Comput. Trends Technol.* **2014**, *11*, 94–98. [CrossRef]
23. Rahman, M.; Afroz, A. Comparison of various classification techniques using different data mining tools for diabetes diagnosis. *J. Softw. Eng. Appl.* **2013**, *6*, 85. [CrossRef]
24. Su, C.; Yang, C.; Hsu, K.; Chiu, W. Data mining for the diagnosis of type II diabetes from three-dimensional body surface anthropometrical scanning data. *Comput. Math. Appl.* **2006**, *51*, 1075–1092. [CrossRef]
25. Firdaus, M.; Nadia, R.; Tama, B. Detecting major disease in public hospital using ensemble techniques. In Proceedings of the IEEE International Symposium on Technology Management and Emerging Technologies (ISTMET), Bandung, Indonesia, 27–29 May 2014; pp. 149–152.
26. Zolfaghari, R. Diagnosis of diabetes in female population of Pima Indian heritage with ensemble of BP neural network and SVM. *Int. J. Comput. Eng. Manag.* **2012**, *15*, 2230–7893.
27. Lee, C. A fuzzy expert system for diabetes decision support application. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2011**, *41*, 139–153. [PubMed]
28. Christobel, Y.; SivaPrakasam, P. The negative impact of missing value imputation in classification of diabetes dataset and solution for improvement. *IOSR J. Comput. Eng. (IOSRJCE)* **2012**, *7*, 5.
29. Nirmala Devi, M.; Appavu, S.; Swathi, U. An amalgam KNN to predict diabetes mellitus. In Proceedings of the IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), Tirunelveli, India, 25–26 March 2013; pp. 691–695.
30. Aslam, M.; Zhu, Z.; Nandi, A.K. Feature generation using genetic programming with comparative partner selection for diabetes classification. *Expert Syst. Appl.* **2013**, *40*, 5402–5412. [CrossRef]
31. Stahl, F.; Johansson, R.; Renard, E. *Ensemble Glucose Prediction in Insulin-Dependent Diabetes. Data Driven Modeling for Diabetes*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 37–71.
32. Gandhi, K.; Prajapati, N.B. Diabetes prediction using feature selection and classification. *Int. J. Adv. Eng. Res. Dev.* **2014**, *1*, 1–7.
33. Varma, K.; Rao, A.; Lakshmi, T.; Rao, P. A computational intelligence approach for a better diagnosis of diabetic patients. *Comput. Electr. Eng.* **2014**, *4*, 1758–1765. [CrossRef]
34. Polat, K.; Güneş, S.; Arslan, A. A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine. *Expert Syst. Appl.* **2008**, *34*, 482–487. [CrossRef]
35. Beloufa, F.; Chikh, M. Design of fuzzy classifier for diabetes disease using modified artificial bee colony algorithm. *Comput. Methods Prog. Biomed.* **2013**, *112*, 92–103. [CrossRef]
36. Chikh, M.; Saidi, M.; Settouti, N. Diagnosis of diabetes diseases using an artificial immune recognition system2 (airs2) with fuzzy k-nearest neighbor. *J. Med. Syst.* **2012**, *36*, 2721–2729. [CrossRef]
37. Sahebi, H.; Ebrahimi, S.; Ashtian, I. Afuzzy classifier based on modified particle swarm optimization for diabetes disease diagnosis. *Adv. Comput. Sci. Int. J.* **2015**, *4*, 11–17.
38. Cheruku, R.; Edla, D.; Kuppili, V. SM-RuleMiner: Spider monkey based rule miner using novel fitness function for diabetes classification. *Comput. Biol. Med.* **2017**, *81*, 79–92. [CrossRef]
39. Tama, B.; Fitri, R. Hermansyah: An early detection method of type-2 diabetes mellitus in public hospital. TELKOMNIKA. *Telecommun. Comput. Electr. Control.* **2013**, *9*, 287–294.
40. Ali, R.; Siddiqi, M.; Idris, M.; Kang, B.; Lee, S. Prediction of diabetes mellitus based on boosting ensemble modeling. In Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence, Belfast, UK, 2–5 December 2014; Springer: Cham, Switzerland, 2014; pp. 25–28.
41. Tama, B.; Rhee, K. Tree-based classifier ensembles for early detection method of diabetes: An exploratory study. *Artif. Intell. Rev.* **2019**, *51*, 355–370. [CrossRef]

42. Bashir, S.; Qamar, U.; Khan, F.; Naseem, L. HMV: A medical decision support framework using multi-layer classifiers for disease prediction. *J. Comput. Sci.* **2016**, *13*, 10–25. [CrossRef]
43. El-Baz, A.; Hassanien, A.; Schaefer, G. Identification of diabetes disease using committees of neural network-based classifiers. In *Machine Intelligence and Big Data in Industry*; Springer: Cham, Switzerland, 2016; pp. 65–74.
44. Junior, J.; Nicoletti, M. An iterative boosting-based ensemble for streaming data classification. *Inf. Fusion* **2019**, *45*, 66–78. [CrossRef]
45. Saleh, E.; Błaszczyński, J.; Moreno, A.; Valls, A.; Romero-Aroca, P.; de la Riva-Fernández, S.; Słowiński, R. Learning ensemble classifiers for diabetic retinopathy assessment. *Artif. Intell. Med.* **2018**, *85*, 50–63. [CrossRef]
46. Nannia, L.; Luminib, A.; Zaffonato, N. Ensemble based on static classifier selection for automated diagnosisof Mild Cognitive Impairment. *J. Neurosci. Methods* **2018**, *302*, 42–46. [CrossRef]
47. Nguyen, T.; Nguyen, M.; Pham, X.; Liew, A. Heterogeneous classifier ensemble with fuzzy rule-based meta learner. *Inf. Sci.* **2018**, *422*, 144–160. [CrossRef]
48. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. *ICML* **1996**, *96*, 148–156.
49. Dwivedi, A. Analysis of computational intelligence techniques for diabetes mellitus prediction. *Neural Comput. Appl.* **2018**, *30*, 3837–3845. [CrossRef]
50. El-Sappagh, S.; Ali, F. DDO: A diabetes mellitus diagnosis ontology. *Appl. Inform.* **2016**, *3*, 5. [CrossRef]
51. Kotsiantis, S. Supervised machine learning: A review of classification techniques. *Informatica* **2007**, *31*, 249–268.
52. Corinna, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.
53. Basheer, I.; Hajmeer, M. Artificial neural networks: Fundamentals, computing, design, and application. *J. Microbiol. Meth.* **2000**, *43*, 3–31. [CrossRef]
54. Ho, T. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844.
55. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
56. Kang, S.; Cho, S.; Kang, P. Multi-class classification via heterogeneous ensemble of one-class classifiers. *Eng. Appl. Artif. Intell.* **2015**, *43*, 35–43. [CrossRef]
57. Moretti, F.; Pizzuti, S.; Panzieri, S.; Annunziato, M. Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling. *Neurocomputing* **2015**, *167*, 3–7. [CrossRef]
58. Kim, M.; Kang, D.; Kim, H. Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. *Expert Syst. Appl.* **2015**, *42*, 1074–1082. [CrossRef]
59. Witten, I.; Frank, E.; Hall, M.; Pal, C. *Data Mining Practical Machine Learning Tools and Techniques*, 4th ed.; Elsevier: Burlington, MA, USA, 2017.
60. Canadian Diabetes Association Clinical Practice Guidelines Expert Committee. Pharmacologic Management of Type 2 Diabetes. *Can. J. Diabetes* **2013**, *37*, S61–S68. [CrossRef] [PubMed]
61. American Diabetes Association. Standards of medical care in diabetes. *Diabetes Care* **2017**, *40* (Suppl. 1), S1–S2. [CrossRef]
62. Almuhaideb, S.; Menai, M. Impact of preprocessing on medical data classification. *Front. Comput. Sci.* **2016**, *10*, 1082–1102. [CrossRef]
63. Fayyad, U.; Irani, K. Multi-interval discretization of continuous valued attributes for classification learning. In Proceedings of the Thirteenth International Joint Conference on Articial Intelligence, Chambéry, France, 28 August–3 September 1993; pp. 1022–1027.
64. Bramer, M. *Principles of Data Mining*, 2nd ed.; Springer: London, UK, 2013.
65. Hall, M.; Holmes, G. Benchmarking Attribute Selection Techniques for Discrete Class Data Mining. *IEEE Trans. Knowl. Data Eng.* **2003**, *15*, 1437–1447. [CrossRef]
66. Brown, G.; Kuncheva, L. *"Good" and "Bad" Diversity in Majority Vote Ensembles, Multiple Classifier Systems*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 124–133.
67. Díez-Pastor, J.; Rodríguez, J.; García-Osorio, C.; Kuncheva, L. Random balance: Ensembles of variable priors classifiers for imbalanced data. *Knowl.-Based Syst.* **2015**, *85*, 96–111. [CrossRef]
68. King, M.; Abrahams, A.; Ragsdale, C. Ensemble learning methods for payper-click campaign management. *Expert Syst. Appl.* **2015**, *42*, 4818–4829. [CrossRef]

69. Majid, A.; Ali, S.; Iqbal, M.; Kausar, N. Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines. *Comput. Methods Programs Biomed.* **2014**, *113*, 792–808. [CrossRef] [PubMed]
70. Matthews, B. Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochim. *Biophys. Acta-Protein Struct.* **1975**, *405*, 442–451. [CrossRef]
71. Kubat, M.; Matwin, S. Addressing the Curse of Imbalanced Training Set: One-Sided Selection. In Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, USA, 8–12 July 1997; pp. 179–186.
72. Ani, R.; Krishna, S.; Anju, N.; Aslam, M.; Deepa, O. IoT Based Patient Monitoring and Diagnostic Prediction Tool using Ensemble Classifier. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1588–1593.