

## Article

# Combining Supervised and Unsupervised Fuzzy Learning Algorithms for Robust Diabetes Diagnosis

Kwang Baek Kim <sup>1,\*</sup>, Hyun Jun Park <sup>2</sup> and Doo Heon Song <sup>3</sup><sup>1</sup> Department of Artificial Intelligence, Silla University, Busan 46958, Republic of Korea<sup>2</sup> Division of Software Convergence, Cheongju University, Cheongju 28503, Republic of Korea<sup>3</sup> Department of Computer Games, Yong-In Art & Science University, Yongin 17145, Republic of Korea

\* Correspondence: gbkim@silla.ac.kr

**Abstract:** In domains that have complex data characteristics and/or noisy data, any single supervised learning algorithm tends to suffer from overfitting. One way to mitigate this problem is to combine unsupervised learning component as a front end of the main supervised learner. In this paper, we propose a hierarchical combination of fuzzy C-means clustering component and fuzzy max–min neural network supervised learner for that purpose. The proposed method is evaluated in a noisy domain (Pima Indian Diabetes open database). The proposed combination showed superior result to standalone fuzzy max–min and backpropagation-based neural network. The proposed method also showed better performance than any single supervised learner tested in the same domain in the literature with high accuracy (80.96%) and was at least competitive in other measures such as sensitivity, specificity, and F1 measure.

**Keywords:** fuzzy max–min neural network; fuzzy c-means; diabetes; supervised learning; unsupervised learning



**Citation:** Kim, K.B.; Park, H.J.; Song, D.H. Combining Supervised and Unsupervised Fuzzy Learning Algorithms for Robust Diabetes Diagnosis. *Appl. Sci.* **2023**, *13*, 351. <https://doi.org/10.3390/app13010351>

Academic Editor: Xiaoyang Liu

Received: 5 November 2022

Revised: 4 December 2022

Accepted: 23 December 2022

Published: 27 December 2022



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Ensemble learning method is designed to compensate the weakness of traditional machine learning algorithms in dealing with complex data [1]. Typically, two or more independent classifiers, either homogeneous or heterogeneous, are executed in parallel and the learning result of each classifier over the same input is combined by some voting mechanisms. Ensemble learning is better than its single best base learning algorithm especially in dealing with high-dimensional, imbalanced, noisy data. Such better generalization power of ensemble may come from the information insufficiency of the training data for single best learner and/or imperfect search process of a single learner [2].

There is another approach to overcome the limitation of a single supervised machine learning algorithm. In general, supervised learning algorithm shows better accuracy than unsupervised learning algorithms since it has a reliable known label (class) in training phase that will guide efficient search process. However, it is also well known that the supervised learner may have more serious overfitting problem that refers to the incident that the supervised learner performs perfectly on training set, only to fail on the testing set or unseen data [3]. One of the main sources of such overfitting is the imperfectness of given features and too strong bias of feature information provided to the input of the supervised learner.

There have been good approaches to combine unsupervised learner and supervised learner to solve such problem. In most cases, the unsupervised learner (component) finds the cluster relations of input data and provide such information to the supervised component that learns the desired patterns between clusters and classes. Such hierarchical combination of unsupervised and supervised learner provides better chance of incremental learning and overlapped clustering result in unsupervised learning phase [4].

For example, in human activity recognition problem, Predict and Cluster method [5] applies clustering algorithm to gather similar activities into the same cluster and different activities into separate clusters. Then, a method based on an encoder/decoder recurrent neural network, can effectively learn distinct actions. It outperforms other single supervised methods in this domain. Recently, for the same domain, a system utilizes K-means first and generate a hyper label that becomes the input set for supervised learner Spatial-Temporal Graph Convolutional Network (ST-GCN) [6]. This approach obtained 9% better result in accuracy than the base supervised algorithm ST-GCN [7] and gives robustness when facing with a new activity by unsupervised component since it can only retain the cluster to which the activity is assigned.

Similar efforts to combine unsupervised learning result with supervised learning component shows plausible result in facies prediction problem [8,9], malware detection [10], land price prediction [11], credit risk assessment [12], miRNA target prediction [13], text summarization tasks [14,15] and image segmentation task [16]. For those systems, usually there is no restrictions on unsupervised algorithm component, that is, any clustering algorithm can be applied and Support Vector Machine (SVM) [17] is frequently used for supervised learning component [8–10,12,13] but standard neural network [11] and decision tree [16] can be applied as well.

Considering that the role of unsupervised learning component is to provide more informative/independent features to the input of the supervised classification component in this hierarchical combination, we can extend the relationship between the input data and output clusters of the unsupervised learning component. If we apply fuzzy clustering algorithms to the unsupervised learning component, that fuzzy clusters maintain the relationship between the cluster and all input with its membership function so that the entire system can maintain more stable and robust relationship between the input data and the final output of the supervised learner. That is, fuzzy clusters maintain multiple associations with input data and the output of unsupervised learning component has overlapped concept from the input data. If such fuzzy clustering component is well combined with the supervised classifier, the entire system might be more flexible, stable, and robust than the standalone supervised learner.

There exist only a few studies on that direction. In [18], the SVM classifier generates a spectral-based classification map, whereas Fuzzy C-Means (FCM) is adopted to provide an ensemble of segmentation maps. In traffic flow prediction problem [19,20], type-2 FCM model classifies the historical data into different traffic flow patterns, and its result becomes the input of the neural network back propagation principle. It is found that the original 21 detectors on the highway have significant correlation among them and FCM based unsupervised learning component extends the mutual independence of the input of neural network (fuzzy clusters) so that the final prediction performance is revealed as better than any type of single neural network classifiers (neural network, recurrent neural network (RNN), and long short-term memory (LSTM)).

In this paper, we propose a hierarchical combination method between fuzzy supervised and fuzzy unsupervised learning algorithm. For unsupervised learning component, we choose most popular standard FCM as other approaches, but we adopt fuzzy max–min neural network (FMM) [21] as the supervised classifier. FMM constructs an N-dimensional hyperbox defined by its Max and Min point. In classification, hyperbox fuzzy sets are aggregated to form a single fuzzy set class. Learning in FMM consists of creating and expanding/contracting hyperboxes in a pattern space. If well-designed, FMM has online learning ability in one-pass through the data samples, and can overcome overlapped classes, and having nonlinear separability [22].

To demonstrate the effect of such hierarchical combination of FCM and FMM, we chose the diabetes prediction problem, which has multiple open databases. Diabetes is a chronic metabolic disease due to a high level of sugar in the blood over a lengthy period. Although it has severe complications, early detection of the disease can reduce the risk factor and severity of the disease. However, the robust and accurate prediction of diabetes

is challenging since there are significant outliers/missing values in the diabetes datasets with limited number of known labeled data [23].

### 2. Materials and Methods

The proposed method for combining fuzzy unsupervised and fuzzy supervised learning algorithm can be demonstrated as Figure 1.

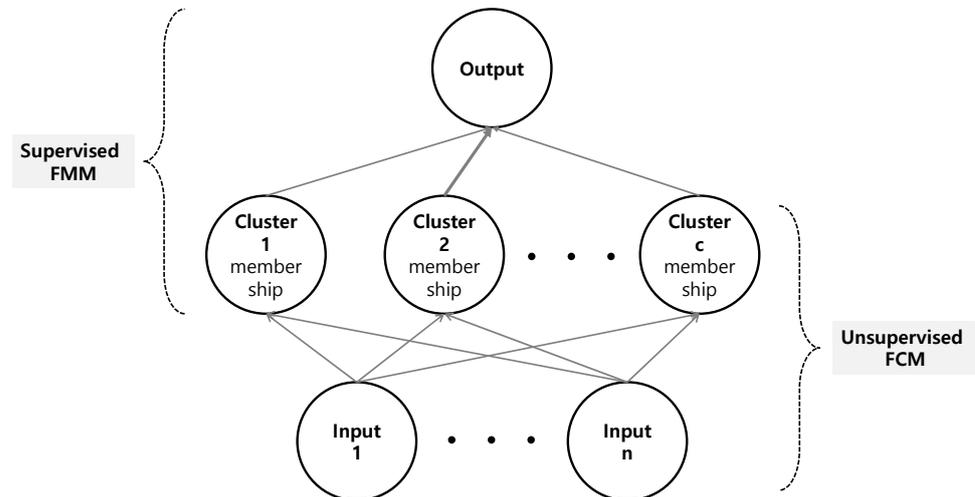


Figure 1. Structure of Proposed Method.

The core input (diabetes patient information) is given to the unsupervised learner FCM. For  $n$  data and  $c$  fuzzy clusters ( $c < n$ ), FCM allows each value to belong to one or more clusters depending on the degree of membership to each cluster.

After FCM process, only non-zero membership clusters act as the input of fuzzy supervised learner (FMM) thus the middle layer and the output layer may not be fully connected as shown in cluster 1 of Figure 1.

The FCM algorithm used in this paper is as Algorithm 1 below.

---

**Algorithm 1.** Fuzzy  $c$ -means process.

---

Step 1. Initialize the number of cluster  $c$  ( $2 \leq c < n$ ), exponential weight  $m = 2$  ( $1 \leq m < \infty$ ), and stopping condition  $\epsilon = 0.0001$

Step 2. Compute the distance between the  $i$ -th cluster centroid  $v_i$  and  $k$ -th data  $x_k$  as Equation (1).

$$d_{ik} = \sqrt{(x_k - v_i)^{\frac{2}{m-1}}} \tag{1}$$

Step 3. Compute the membership degree of  $k$ -th data and  $i$ -th cluster as Equation (2).

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{jk}}{d_{ik}}\right)^{\frac{2}{m-1}}} \tag{2}$$

Step 4. Compute centroid of each cluster by Equation (3).

$$v_i = \frac{\sum_{k=1}^n (u_{ik})^2 x_k}{\sum_{k=1}^n (u_{ik})^2} \tag{3}$$

Step 5. Stop the process if the difference between the centroid position by Equation (3) and previous centroid is less than  $\epsilon$ , otherwise, go to Step 2 and repeat the process.

---

The FMM learning process shown in Algorithm 2 consists of Initialization, Expansion, Overlap Test, and Contraction [21]. It selects an input pattern and find the closest hyperbox to that pattern and expand. If a hyperbox cannot satisfy the expansion criteria, a new hyperbox is formed and added to the system. Hyperbox overlap may cause undesirable ambiguity; therefore, a contraction process is used to eliminate any undesired hyperbox

overlaps. In classification learning, the overlap is eliminated only for hyperboxes that represent different classes.

---

**Algorithm 2.** Fuzzy max–min neural network learning process.

---

$w_{hg}$ : weight,  $\theta_h$ : bias term,  $p$ : number of patterns,  $\alpha$ : learning rate,  
 $\eta$ : momentum,  $x_g$ : input pattern,  $t_h$ : target value,  $y_h$ : output

Step 1. Set input  $x_g$  (where  $g = 1, \dots, d$ ) and target value  $t_h$  (where  $h = 1, \dots, z$ ).

Step 2. Initialize  $w_{hg}$  and  $\theta_h$ .

Step 3. Compute FMM NET as Equation (4).

$$\text{NET} = \bigvee_{h=1}^z \left\{ \bigwedge_{g=1}^d \{x_g, w_{hg}\} \right\} \tag{4}$$

where  $\vee$  is fuzzy OR operator and  $\wedge$  is fuzzy AND operator.

Step 4. Compute the output  $y_j$  as Equation (5).

$$y_h = \text{NET} \vee \theta_h \tag{5}$$

Step 5. Compute error term TSS as Equation (6).

$$TSS = \frac{1}{2} \sum_{p=1}^l \sum_{h=1}^z (t_h^p - y_h^p)^2. \tag{6}$$

where  $p$  denotes the number of patterns.

Step 6. Update weight  $w_{hg}$  and bias  $\theta_h$  as Equation (7).

$$\begin{aligned} \Delta w_{hg}(r+1) &= \Delta w_{hg}(r) + \frac{\partial y_h}{\partial w_{hg}}(t_h - y_h) \\ \Delta \theta_h(r+1) &= \Delta \theta_h(r) + \frac{\partial y_h}{\partial \theta_h}(t_h - y_h) \\ \frac{\partial y_h}{\partial w_{hg}} &= \begin{cases} 1, & y_h = w_{hg} \\ 0, & \text{otherwise.} \end{cases} \\ \frac{\partial y_h}{\partial \theta_h} &= \begin{cases} 1, & y_h = \theta_h \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{7}$$

$$w_{hg}(r+1) = w_{hg}(r) + \alpha \Delta w_{hg}(r+1) + \eta \Delta w_{hg}(r)$$

$$\theta_h(r+1) = \theta_h(r) + \alpha \Delta \theta_h(r) + \eta \Delta \theta_h(r)$$

Step 7. UStop if TSS is less than predefined threshold.

Otherwise, go to Step 2.

---

Figure 2 on the next page shows a flow diagram of the proposed FCM–FMM combination process.

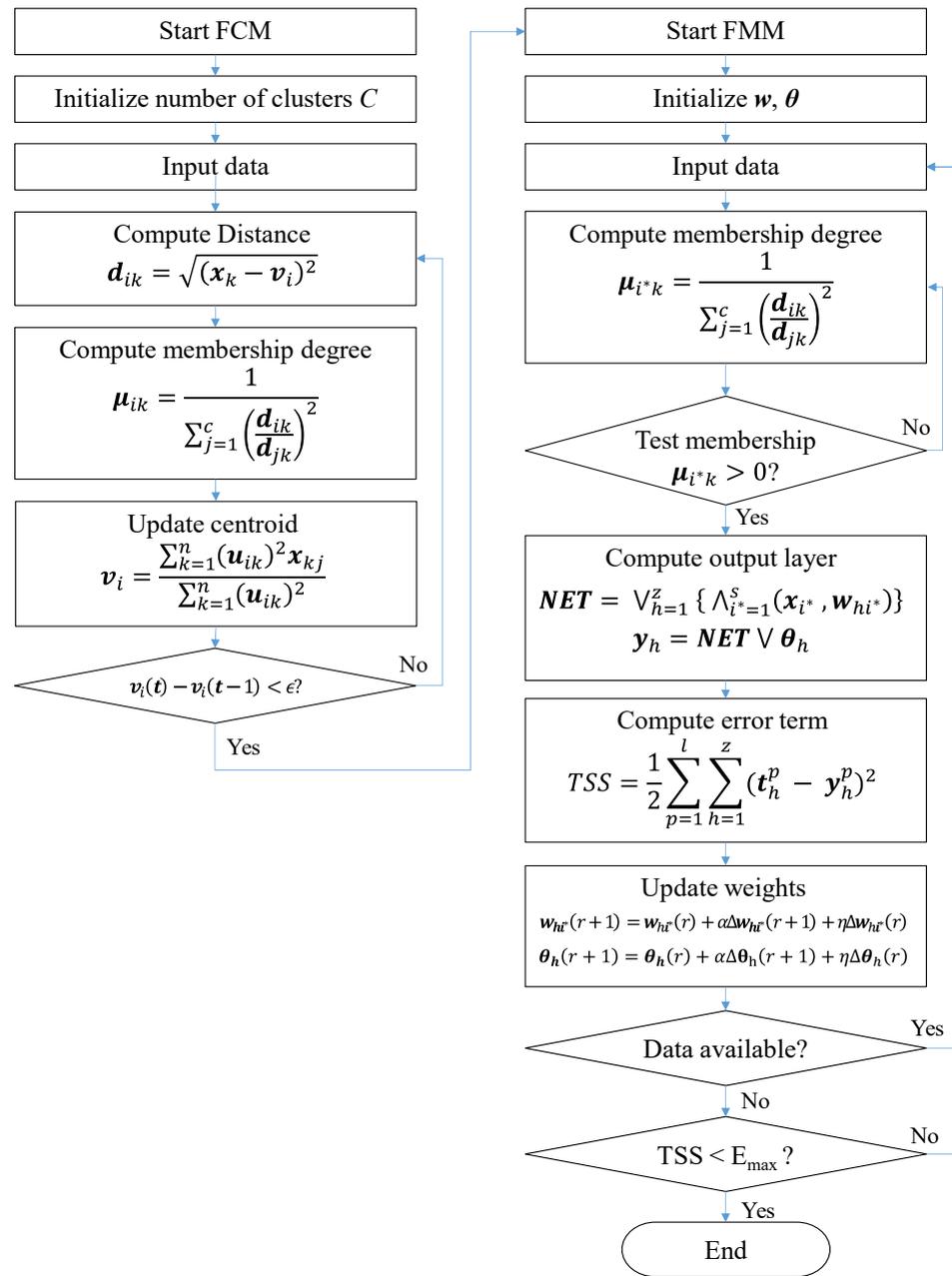


Figure 2. Overall FCM–FMM combination process.

### 3. Results

The proposed method is implemented using Visual Studio 2019 C# with AMD Ryzen 3 3300X 4-Core Processor, 3.79 GHz and 16 GB RAM PC.

#### 3.1. Dataset

We used open database <https://www.kaggle.com/datasets/mathchi/diabetes-dataset> (“Diabetes Dataset”, Kaggle, last modified 2020, accessed on 1 July 2022) in this paper. This Pima Indian Diabetes (PID) dataset, containing 768 female diabetic patients’ information who are at least 21 years old of Pima Indian heritage. is originally from the National Institute of Diabetes and Digestive and Kidney Diseases [24] and only 268 patients (34.9%) are classified into “having diabetes” among them.

This dataset has eight attributes and one binary class attribute (class value 1 is “diabetes”) with significant number of missing values in several attributes. We use “fill-in-the-

average" strategy for missing value treatment, and all attributes are normalized since all attributes are real values. Attributes and number of missing values for each attribute are summarized in Table 1.

**Table 1.** PID dataset configuration.

Attributes	Missing
Number of times pregnant	0
Plasma glucose concentration	5
Diastolic blood pressure	35
Triceps skin fold thickness	221
2-hour serum insulin	375
Body mass index	11
Diabetes pedigree function	0
Age	0
Class (0 or 1)	0

Diabetes Pedigree Function (DPF) is defined to provide a synthesis of the diabetes mellitus history in relatives and the genetic relationship of those relatives to the subject [24].

### 3.2. Performance Evaluation

We use random stratified sampled hold-out to setup the training set and the test set as 70% of the dataset is used in the training phase. The experiment is repeated 30 times and the dataset is randomly sampled at each hold-out cycle. The performance result reported here is the average of these 30 times hold-out experiments.

The effect of the proposed method is measured with *sensitivity*, *specificity*, *accuracy*, and *F1 score* in comparison with the backpropagation neural network (BPNN) with the unipolar sigmoid function and stand-alone FMM. All performance measures are based on the quantity of *TP* (True positive), *TN* (True Negative), *FP* (False Positive), and *FN* (False Negative) as shown in Table 2.

**Table 2.** Performance evaluation metrics.

<i>Accuracy</i>	$\frac{(TP + TN)}{(TP + TN + FP + FN)}$
<i>Sensitivity</i>	$\frac{TP}{TP + FN}$
<i>Specificity</i>	$\frac{TN}{TN + FP}$
<i>Precision</i>	$\frac{TP}{TP + FP}$
<i>F1 Score</i>	$2 \left( \frac{Precision * Sensitivity}{Precision + Sensitivity} \right)$

The effect of combining fuzzy unsupervised learner and fuzzy supervised learner is summarized in Table 3. A simple fast supervised learner FMM is extremely good in *sensitivity* but extremely poor in *specificity* in this experiment. That means FMM alone produces too many *FPs*. By combining FCM and FMM, the proposed method shows more stable and robust result with highest *F1 score* and *accuracy*. Standard BPNN is not remarkable in all measures.

**Table 3.** Performance summary.

Algorithm	<i>Sensitivity</i>	<i>Specificity</i>	<i>Accuracy</i>	<i>F1 Score</i>
FCM–FMM	78.79	70.27	80.96	74.16
FMM	94.67	49.88	64.91	65.29
BPNN	60.88	65.23	74.96	62.78

To verify the performance of three tested algorithms statistically, we use one-way ANOVA (Analysis of Variance) test and subsequent Tukey test to see if there exist any statistically significant group mean differences for each performance metric. We use open web site for statistical computation (<http://vassarstats.net/> (accessed on 4 November 2022)) for this computation and the test results are summarized as shown in Tables 4–7 as follows: In Tables 4–7, HSD is the absolute [unsigned] difference between any two-sample means required for significance at the designated level (HSD [0.05] for the 0.05 level), M1 is the proposed FCM-FMM, M2 is Standalone FMM, and M3 is BPNN.

**Table 4.** ANOVA and Tukey Test for *Sensitivity*.

Source	SS	df	MS	F	<i>p</i>
<b>Treatment</b>	1.7149	2	0.8575	389.06	<0.0001
<b>Error</b>	0.1917	89	0.0022		
<b>Total</b>	1.9066	91			
HSD [0.05] = 0.03; HSD [0.01] = 0.04					
M1 vs. M2 $p < 0.01$ , M1 vs. M3 $p < 0.01$ , M2 vs. M3 $p < 0.01$					

**Table 5.** ANOVA and Tukey Test for *Specificity*.

Source	SS	df	MS	F	<i>p</i>
<b>Treatment</b>	0.6764	2	0.3382	298.39	<0.0001
<b>Error</b>	0.0986	89			
<b>Total</b>	0.775	91			
HSD [0.05] = 0.02; HSD [0.01] = 0.03					
M1 vs. M2 $p < 0.01$ , M1 vs. M3 $p < 0.01$ , M2 vs. M3 $p < 0.01$					

**Table 6.** ANOVA and Tukey Test for *Accuracy*.

Source	SS	df	MS	F	<i>p</i>
<b>Treatment</b>	0.5223	2	0.261148	49.53	<0.0001
<b>Error</b>	0.464	89	0.005272		
<b>Total</b>	0.9863	91			
HSD [0.05] = 0.04; HSD [0.01] = 0.06					
M1 vs. M2 $p < 0.01$ , M1 vs. M3 $p < 0.05$ , M2 vs. M3 $p < 0.01$					

**Table 7.** ANOVA and Tukey Test for *F1 Score*.

Source	SS	df	MS	F	<i>p</i>
<b>Treatment</b>	0.3042	2	0.1521	15.52	<0.0001
<b>Error</b>	0.8723	89	0.0098		
<b>Total</b>	1.1765	91			
HSD [0.05] = 0.06; HSD [0.01] = 0.08					
M1 vs. M2 $p < 0.01$ , M1 vs. M3 $p < 0.01$ , M2 vs. M3 nonsignificant					

All four metrics have significant ( $p < 0.0001$ ) in ANOVA test and all three compared algorithms showed significant mean difference for all measures, but FMM and BPNN have no significant difference in *F1 score* metric (Table 7).

The PID Dataset is used in many other machine learning algorithms and their performance is reported in the literature. However, Hasan et al.'s approach [23] applied outlier rejection preprocessing so that the number of actual instances learned is different and Alarm et al. approach [25] is designed to find optimum number of hidden layers in neural network model thus the purpose is different and only the *accuracy* (75.7%) is comparable to our result.

Two previous works reported the same measurement as ours and no specific preprocessing is treated. Thus, we report the comparison of our proposed method with other single supervised learning algorithms from the literature in Table 8.

**Table 8.** Performance comparison summary from the literature.

Algorithm	Sensitivity	Specificity	Accuracy	F1 Score
Naïve Bayes [26]	75.9	76.3	76.3	76.0
SVM [26]	42.4	65.1	65.1	51.3
Decision Tree [26]	73.5	73.8	73.8	73.6
Soft Voting [27]	70.0	73.1	79.1	71.6
FCM–FMM	78.8	70.3	81.0	74.2

With the same experimental condition, our proposed method is the best in *sensitivity* and the *accuracy* and the second in *F1 score*. Thus, at least, this hierarchical combination of rather simple fuzzy unsupervised and supervised learner is competitive in this highly vague and complex domain. Naive Bayes [28] is the best for other two measures (*specificity* and *F1 score*) among five compared algorithms in Table 8.

#### 4. Discussion

In supervised learning, model overfitting is a hurdle to make the learner to be robust and stable over unseen data. Over-fitted supervised learners tend to memorize more than necessary including noise on the training set, instead of learning the discipline hidden behind the data [3]. Statistically, overfitting occurs when we use a model that is more flexible than it needs to be, or the model includes irrelevant components [29]. Noise, missing values, insufficient information representation of the input, and unnecessary input parameter correlation can be the cause of such undesirable phenomenon in supervised learning.

Combining unsupervised learning components as a preprocessing unit of the supervised learning component may mitigate such model overfitting problem since the result of unsupervised learning (clusters) can represent an intermediate concept or noise filtering scheme for the supervised learner. Such attempts have shown encouraging result in many complex machine learning problems that have high uncertainty and noisy data in engineering and management [5–16].

In this paper, we propose a hierarchical combination of fuzzy unsupervised learning component (FCM) and supervised learning component (FMM: Fuzzy Max–min Neural Network) in diabetes diagnosis that is known as a highly noisy domain since it has not much labelled data and has many missing values in the dataset. We expect that clusters produced by the FCM would give more mutually independent input to the supervised learning component (FMM) so that the performance of the proposed approach can be more robust than the standalone supervised learner.

In experiment using the 30-fold 7:3 random stratified holdout procedure, we report that the proposed FCM–FMM combination is better than standard back propagation-based neural network and standalone FMM as expected in all measures except the *sensitivity*. FMM that shows high sensitivity is extremely poor in *specificity* in that it generates too many false positives.

The performance of our proposed FCM–FMM combination is compared with various supervised algorithms using the same PID database and the same measures (*sensitivity*, *specificity*, *accuracy*, *F1 score*) reported in other studies. FCM–FMM is the best in sensitivity and the accuracy and the second in F1 score and Naive Bayes is the best in other two mea-

tures among five algorithms thus we can conclude that this fuzzy hierarchical combination is at least competitive in the highly noisy problem domain such as Diabetes diagnosis.

However, the limitation of this work should be noted that the performance of the proposed algorithm is tested on the single open dataset (PID database). Although the PID dataset chosen in this paper is a good example of testing the robustness of the learning algorithms under fuzzy insufficient input information with many missing values, any direct generalization of this experiment result may not be appropriate to judge the overall performance of tested algorithms. The proposed method should be tested for many other noisy datasets to prove its efficacy in the future. Additionally, in model validation, we used repeated hold-out for simplicity with the expense of pessimistic bias of the performance partly because two studies we compared in Table 8 took different policies (10-fold cross validation in [26] and repeated holdout in [27]). No further resampling technique is applied to our method to refine the performance metric to maintain fair comparison in Table 8.

Further study is also necessary for achieving better accuracy and robustness in this combination such as outlier rejection, kernel FCM, or applying other variants of FMM summarized in [22].

**Author Contributions:** Conceptualization, K.B.K. and D.H.S.; methodology, K.B.K.; software, K.B.K. and H.J.P.; analysis, K.B.K. and D.H.S.; resources, K.B.K.; data curation, K.B.K.; writing—original draft preparation, K.B.K., D.H.S. and H.J.P.; writing—review and editing, K.B.K., D.H.S. and H.J.P.; visualization, D.H.S. and H.J.P.; supervision, K.B.K. and H.J.P.; project administration, K.B.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** Following are results of a study on the “Leaders in INdustry-university Cooperation 3.0” Project, supported by the Ministry of Education and National Research Foundation of Korea.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to Institutional regulations.

**Conflicts of Interest:** The authors declare no conflict of interest regarding the publication of this paper.

## References

1. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **2020**, *14*, 241–258.
2. Polikar, R. Ensemble Learning. In *Ensemble Machine Learning*; Springer: Boston, MA, USA, 2012; pp. 1–34.
3. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [[CrossRef](#)]
4. Lee, H.M.; Lai, C.S. Supervised extended ART: A fast neural network classifier trained by combining supervised and unsupervised learning. *Appl. Intell.* **1996**, *6*, 117–128. [[CrossRef](#)]
5. Kun, S.; Xiulong, L.; Eli, S. PREDICT & CLUSTER: Unsupervised Skeleton Based Action Recognition. *arXiv* **2019**, arXiv:1911.12409.
6. Budisteanu, E.A.; Mocanu, I.G. Combining Supervised and Unsupervised Learning Algorithms for Human Activity Recognition. *Sensors* **2021**, *21*, 6309. [[CrossRef](#)]
7. Yan, S.; Xiong, Y.; Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *arXiv* **2018**, arXiv:1801.07455. [[CrossRef](#)]
8. Ippolito, M.; Ferguson, J.; Jenson, F. Improving facies prediction by combining supervised and unsupervised learning methods. *J. Pet. Sci. Eng.* **2021**, *200*, 108300. [[CrossRef](#)]
9. Fadokun, D.O.; Oshilike, I.B.; Onyekonwu, M.O. Supervised and Unsupervised Machine Learning Approach in Facies Prediction. In Proceedings of the SPE Nigeria Annual International Conference and Exhibition, Virtual, 11 August 2020.
10. Comar, P.M.; Liu, L.; Saha, S.; Tan, P.N.; Nucci, A. Combining Supervised and Unsupervised Learning for Zero-Day Malware Detection. In Proceedings of the 2013 IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 2022–2030.
11. Lee, C. Predicting land prices and measuring uncertainty by combining supervised and unsupervised learning. *Int. J. Strateg. Prop. Manag.* **2021**, *25*, 169–178. [[CrossRef](#)]
12. Bao, W.; Lianju, N.; Yue, K. Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. *Expert Syst. Appl.* **2019**, *128*, 301–315. [[CrossRef](#)]
13. Sedaghat, N.; Fathy, M.; Modarressi, M.H.; Shojaie, A. Combining supervised and unsupervised learning for improved miRNA target prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2017**, *15*, 1594–1604. [[CrossRef](#)]
14. Mao, X.; Yang, H.; Huang, S.; Liu, Y.; Li, R. Extractive summarization using supervised and unsupervised learning. *Expert Syst. Appl.* **2019**, *133*, 173–181. [[CrossRef](#)]

15. Wong, K.F.; Wu, M.; Li, W. Extractive Summarization Using Supervised and Semi-Supervised Learning. In Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, UK, 18–22 August 2008; pp. 985–992.
16. Hashemzadeh, M.; Azar, B.A. Retinal blood vessel extraction employing effective image features and combination of supervised and unsupervised machine learning methods. *Artif. Intell. Med.* **2019**, *95*, 1–15. [[CrossRef](#)] [[PubMed](#)]
17. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 237–297. [[CrossRef](#)]
18. Alajlan, N.; Bazi, Y.; Melgani, F.; Yager, R.R. Fusion of supervised and unsupervised learning for improved classification of hyperspectral images. *Inf. Sci.* **2012**, *217*, 39–55. [[CrossRef](#)]
19. Tang, J.; Yu, S.; Liu, F.; Chen, X.; Huang, H. A hierarchical prediction model for lane-changes based on combination of fuzzy C-means and adaptive neural network. *Expert Syst. Appl.* **2019**, *130*, 265–275. [[CrossRef](#)]
20. Tang, J.; Li, L.; Hu, Z.; Liu, F. Short-term traffic flow prediction considering spatio-temporal correlation: A hybrid model combining type-2 fuzzy C-means and artificial neural network. *IEEE Access* **2019**, *7*, 101009–101018. [[CrossRef](#)]
21. Simpson, P.K. Fuzzy min-max neural networks. I. Classification. *IEEE Trans. Neural Netw.* **1992**, *3*, 776–786. [[CrossRef](#)]
22. Alhroob, E.; Mohammed, M.F.; Lim, C.P.; Tao, H. A critical review on selected fuzzy min-max neural networks and their significance and challenges in pattern classification. *IEEE Access* **2019**, *7*, 56129–56146. [[CrossRef](#)]
23. Hasan, M.K.; Alam, M.A.; Das, D.; Hossain, E.; Hasan, M. Diabetes prediction using ensembling of different machine learning classifiers. *IEEE Access* **2020**, *8*, 76516–76531. [[CrossRef](#)]
24. Smith, J.W.; Everhart, J.E.; Dickson, W.C.; Knowler, W.C.; Johannes, R.S. Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus. In Proceedings of the Annual Symposium on Computer Application in Medical Care, Washington, DC, USA, 6–9 November 1988; p. 261.
25. Alam, T.M.; Iqbal, M.A.; Ali, Y.; Wahab, A.; Ijaz, S.; Baig, T.I.; Hussain, A.; Malik, M.A.; Raza, M.M.; Ibrar, S.; et al. A model for early prediction of diabetes. *Inform. Med. Unlocked* **2019**, *16*, 100204. [[CrossRef](#)]
26. Sisodia, D.; Sisodia, D.S. Prediction of diabetes using classification algorithms. *Procedia Comput. Sci.* **2018**, *132*, 1578–1585. [[CrossRef](#)]
27. Kumari, S.; Kumar, D.; Mittal, M. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. *Int. J. Cogn. Comput. Eng.* **2021**, *2*, 40–46. [[CrossRef](#)]
28. Webb, G.I.; Boughton, J.R.; Wang, Z. Not so naive bayes: Aggregating one-dependence estimators. *Mach. Learn.* **2005**, *58*, 5–24. [[CrossRef](#)]
29. Hawkins, D.M. The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1–12. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.