

Comparison of Fuzzy Integral-Fuzzy Measure based Ensemble Algorithms with the State-of-the-art Ensemble Algorithms

Utkarsh Agrawal^{1,*}, Anthony J. Pinar² Christian Wagner¹,
Timothy C. Havens^{2,3}, Daniele Soria⁴, and Jonathan M. Garibaldi¹

¹ School of Computer Science, The University of Nottingham, Nottingham, NG8 1BB, UK

² Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA

³ Department of Computer Science, Michigan Technological University, Houghton, MI, USA

⁴ Department of Computer Science, University of Westminster, London, WIW 6UW, UK

*utkarsh.agrawal@nottingham.ac.uk

Abstract. The Fuzzy Integral (FI) is a non-linear aggregation operator which enables the fusion of information from multiple sources in respect to a Fuzzy Measure (FM) which captures the worth of both the individual sources and all their possible combinations. Based on the expected potential of non-linear aggregation offered by the FI, its application to decision-level fusion in ensemble classifiers, i.e. to fuse multiple classifiers outputs towards one superior decision level output, has recently been explored. A key example of such a FI-FM ensemble classification method is the Decision-level Fuzzy Integral Multiple Kernel Learning (DeFIMKL) algorithm, which aggregates the outputs of kernel based classifiers through the use of the Choquet FI with respect to a FM learned through a regularised quadratic programming approach. While the approach has been validated against a number of classifiers based on multiple kernel learning, it has thus far not been compared to the state-of-the-art in ensemble classification. Thus, this paper puts forward a detailed comparison of FI-FM based ensemble methods, specifically the DeFIMKL algorithm, with state-of-the art ensemble methods including Adaboost, Bagging, Random Forest and Majority Voting over 20 public datasets from the UCI machine learning repository. The results on the selected datasets suggest that the FI based ensemble classifier performs both well and efficiently, indicating that it is a viable alternative when selecting ensemble classifiers and indicating that the non-linear fusion of decision level outputs offered by the FI provides expected potential and warrants further study.

Keywords: Ensemble Classification Comparison, Fuzzy Measures, Fuzzy Integrals, Adaboost, Bagging, Majority Voting and Random Forest

1 Introduction

Ensemble classifiers are a set of classification algorithms with the objective of classifying data objects by combining the outcome of each individual classifier,

Table 1: Acronyms and Notation

FM	Fuzzy Measure
FI	Fuzzy Integral
CFI	Choquet Fuzzy Integral
RAV	Recursive Weighted Power Mean Aggregation Operator
SSE	Sum of squared error
SVM	Support Vector Machines
nf	number of features in each dataset
DeFIMKL	Decision-level Fuzzy Integral Multiple Kernel Learning
MJSVM	Majority Voting with Support Vector Machines ensemble classifier
X	set of information sources i.e. $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$
$h(x_i)$	support of the question for the source x_i
g	Fuzzy Measure
$g(A)$	Fuzzy Measure for subset A
$f_k(x)$	output by the k th classifier in the ensemble
$f^g(x)$	final decision by the ensemble using CFI with respect to the FM g

generally using weights. Many combination techniques exist in the literature including Boosting, Bagging, Random Forest, Majority Voting, etc. [1]. These ensemble methods have been very popular in the machine learning community due to their ability of producing more accurate results than individual classifiers [2] in a very wide range of application areas [3].

Another approach to obtain ensemble classification is the use of the Fuzzy Integral (FI) aggregation defined with respect to a Fuzzy Measure (FM) [4–8]. The FI is a non-linear aggregation operator to fuse weighted information from multiple sources, where the weights are captured by a FM. The FM not only captures the worth of the individual sources, but also the weights of all subset of sources. Recently, a FM-FI based ensemble classification algorithm Decision-level Fuzzy Integral Multiple Kernel Learning (DeFIMKL) [4] was introduced, which aggregates the results of kernel-SVMs through the use of Choquet Fuzzy Integral (CFI) with respect to a FM learned by a regularised quadratic programming approach. Upon initial investigation in [4], [9], the accuracy of FI-FM based ensemble classification method were found to be better than classifiers based on multiple kernel learning. These works further concluded that DeFIMKL was the best among decision level fusion based FI-FM based ensemble classifiers and thus it has been selected in the work as representative of the FI-FM based ensemble classifier family. However, no in-depth comparison of the FI-FM based ensemble classifier with other ensemble methods have been found in the literature [4–9] (discussed in detail in section 2.3). Thus, the motivation of this study is to determine the performance of FM-FI based ensemble methods for the purpose of ensemble classification. We therefore compare DeFIMKL (FI-FM based ensemble classifier) with the state-of-the art ensemble methods including Adaboost, Bagging, Random Forest and Majority Voting on 20 datasets from UCI machine learning repository.

We focus on the FI-FM ensembles as FIs are powerful non-linear aggregation functions (unlike most of the ensemble methods which perform linear com-

binations) which are capable of exploiting interactions between the models in the ensemble (through FM). FI-FM based ensembles have found applications in numerous domains including software defect prediction [10], Multi-Criteria Decision Making (MCDM) [11], Brain Computer Interface (BCI) [12], Face Recognition in Computer Vision [13, 14], Forensic science [15], Explosive Hazard Detection [16].

In the following section (Section 2) we discuss the literature on FI-FM based ensemble classification methods. In this section we also discuss the background of Adaboost, Bagging, Majority Voting and Random Forest ensemble classification methods. In Section 3 the UCI datasets are described along with the experimental settings of the selected algorithms. Section 4 presents the results and discussion followed by conclusions and future works in section 5. Table 1 lists the most commonly used acronyms in the paper.

2 Background

2.1 Fuzzy Measure

Fuzzy Measure (FM) captures the worth of each information source and all their possible combinations i.e. every subset in a power set [4], [17].

Let $X = \{x_1, \dots, x_n\}$ be a discrete and finite set of information sources and $g : 2^X \rightarrow [0, 1]$ be a FM having the following properties:

- P1: Boundary condition, i.e., $g(\emptyset) = 0, g(X) = 1$, and
- P2: Monotonic and non-decreasing, i.e., $g(A) \leq g(B) \leq 1$, if $A \subseteq B \subseteq X$.

For an infinite domain X there is an additional property to ensure continuity; however, it is not applicable in this paper as X is finite and discrete. In the context of multi-source data fusion, $g(A)$ represents the weight or importance of subset A . The FM values of the singletons i.e. $g(x_i)$ are commonly called the densities. Three major approaches have been used to determine FMs: a) Experts: FMs could be specified by the experts, although it would be virtually impossible to specify FM for large collection of sources. b) Algorithms: Several algorithmic methods including Sugeno λ -measure and S-decomposable measure have been proposed in the literature [18, 19]. This method needs the weights of the individual sources to be defined in advance i.e. this method builds FMs from given source densities. c) Optimisation: Various methods including evolutionary algorithms and quadratic programming have been used to generate FMs. [17]. FMs derived using optimisation methods have been used in this work (described in Section 2.4) as they extract weights from the training data, where the worth of the sources are not known in advance. The information quantified by these FMs are combined using the aggregation operators, defined in the next subsection.

2.2 Fuzzy Integral

Fuzzy Integrals (FIs) are often used as non-linear aggregation functions which combine information from multiple sources using the worth of each subset of

sources (provided by a FM ‘g’) and the support of the question (the evidence) [4, 17]. In the context of ensemble classifiers, FIs together with FMs extend the concept of weighted average ensembles and are able to capture the interactions among the classifiers in the ensemble, resulting in a non-linear ensemble classifier. The two most commonly used FIs in the literature include Choquet Fuzzy Integral (CFI) and Sugeno Fuzzy Integral (SFI) [20], although in this work CFI is in focus which is defined as follows:

Choquet Fuzzy Integral: Let $h : X \rightarrow [0, \infty)$ be a real valued function that represents the evidence or support of a hypothesis. The discrete Choquet Fuzzy Integral (CFI) [4, 5, 17, 20] can be defined as:

$$\int_{CFI} h \circ g = CFI_g(h) = \sum_{i=1}^n h(x_{\pi(i)})[g(A_i) - g(A_{i-1})] \quad (1)$$

where π is a permutation of X such that $h(x_{\pi(1)}) \geq h(x_{\pi(2)}) \geq \dots \geq h(x_{\pi(n)})$, $A_i = [x_{\pi(1)}, \dots, x_{\pi(i)}]$ and $g(A_0) = 0$. More detail on the property of FIs and the CFI can be found in [21]. The next subsection discusses the literature on FI-FM based ensemble. It also presents the gap and motivation of the current study.

2.3 Related Work

In the past decade, researchers have turned their attention towards FI-FM based ensemble classifiers, and proposed a number of FI-FM ensembles generating FM from fuzzy densities i.e. algorithmic FM [14, 18, 19]. For example, Wang et al. [19] proposed the use of posterior probabilities to obtain the fuzzy densities from the ensemble of heterogeneous classifiers. Subsequently, the λ -measure was used to obtain the FM from the densities followed by aggregation using the CFI. The ensemble model was compared with five individual classifiers on the Satimage dataset. In another work, Fakhar et al. [18] proposed the use of the training accuracy and the fuzzy entropy (the reliability of information provided by each information source) to generate fuzzy densities followed by aggregation using the CFI. The proposed FI-FM based ensemble model was compared with seven fuzzy set theory based fusions, all of which are multibiometric identification systems [22]. The accuracy of the proposed FI-FM based ensemble outperformed the previously used classification models on the NIST database. Similarly, Wang et al [14] proposed a FI-FM ensemble where the fuzzy densities are generated using the accuracy rate, error distance and the failure extent of the Neural Network models. The model was tested on the JAFFE facial expression database and compared with five Neural Network models.

In another set of studies, Anderson et al. [23] proposed the use of Genetic Algorithm (GA) (optimisation method) to learn FMs. This study indicated that FI-FM based ensemble classifiers could learn the FMs from the training dataset, leading to efficient data-driven FMs. Hu et al. [9] extended the work and proposed a Fuzzy Integral-Genetic Algorithm (FIGA) ensemble classifier which aggregates results of SVM classifiers using the CFI w.r.t. FM learned through the hybrid of Sugeno λ -measure and GA. FIGA generated the initial measure through the use of Sugeno λ -measure followed by GA to search for an optimal FM through

the error optimisation. The resultant ensemble was compared with the Multiple Kernel Learning Group Lasso (MKLGL) ensemble classifier on three datasets. FIGA performed better than MKLGL on all the three datasets.

Pinar and colleagues [4–8] built upon the previous works on data-driven FMs and proposed Decision-level Fuzzy Integral Multiple Kernel Learning (DeFIMKL) algorithm as an alternative to algorithmic and algorithm-optimisation hybrid FMs, which aggregates the outputs of SVM classifiers through the use of CFI with respect to a FM learned through a regularised quadratic programming approach. DeFIMKL was compared to FIGA, MKGL and other FI-FM based ensemble classifiers for six datasets. FIGA and DeFIMKL were best among the feature level fusion classifiers and decision level fusion classifiers respectively.

The researchers in all the above works concluded that FI-FM ensemble performed better than individual classifiers, but they left two important questions unanswered. First the comparison of FI-FM based ensemble with other state-of-the-art ensemble methods and secondly the performance on multiple datasets. In this paper we aim to answer these two questions and thus, compare FI-FM ensemble classifiers with other state-of-the-art ensemble classifiers. Since DeFIMKL was best among the FM-FI based ensemble classifier family for decision level fusion, it was selected as the representative of the FI-FM based ensemble classifiers, described in the next subsection.

2.4 Non-linear FM-FI ensemble classifier: DeFIMKL

Let $f_k(\mathbf{x}_i)$ be the normalised output generated by the k th classifier in an ensemble on a feature vector \mathbf{x}_i . The overall decision of the ensemble is computed by the Choquet Integral, where g encodes the relative worth of each classifier in the ensemble. Thus, the output of the ensemble with respect to the FM g on feature-vector \mathbf{x}_i is produced by $f^g(\mathbf{x}_i)$, mathematically described as follows,

$$f^g(\mathbf{x}_i) = \sum_{k=1}^m f_{\pi(k)}(\mathbf{x}_i)[g(A_k) - g(A_{k-1})], \quad (2)$$

where $A_k = f_{\pi(1)}(\mathbf{x}_i), \dots, f_{\pi(k)}(\mathbf{x}_i)$, such that $f_{\pi(1)}(\mathbf{x}_i) \geq f_{\pi(2)}(\mathbf{x}_i) \geq \dots \geq f_{\pi(m)}(\mathbf{x}_i)$. It can be shown that (2) can be reformulated as

$$f^g(\mathbf{x}_i) = \sum_{k=1}^m [f_{\pi(k)}(\mathbf{x}_i) - f_{\pi(k+1)}(\mathbf{x}_i)]g(A_k). \quad (3)$$

where $f_{\pi(m+1)} = 0$. Pinar et al. [4] proposed to learn FM g using a regularised sum of squared error (SSE) optimisation, described as follows,

$$E^2 = \sum_{i=1}^n (f^g(\mathbf{x}_i) - y_i)^2 + v(u), \quad (4)$$

where y_i is the class label for \mathbf{x}_i and $v(u)$ is a regularisation function. Equation (4) can be further expanded as

$$E^2 = \sum_{i=1}^n (H_{x_i}^T * u - y_i)^2 + v(u), \quad (5)$$

where y_i is the actual class label for \mathbf{x}_i , u is lexicographically ordered FM g i.e. $u = (g\{x_1\}, g\{x_2\}, \dots, g\{x_1 \cup x_2\}, g\{x_1 \cup x_3\}, \dots, g\{x_1 \cup x_2 \cup \dots \cup x_m\})$, and

$$H_{x_i} = \begin{pmatrix} \cdot \\ f_{\pi(1)}(\mathbf{x}_i) - f_{\pi(2)}(\mathbf{x}_i) \\ \cdot \\ 0 \\ \cdot \\ \cdot \\ f_{\pi(m)}(\mathbf{x}_i) - 0 \end{pmatrix}, \quad (6)$$

where H_{x_i} is of size $(2^m - 1)$ and contains all the difference terms $f_{\pi(k)}(\mathbf{x}_i) - f_{\pi(k+1)}(\mathbf{x}_i)$ at the corresponding locations of A_k in u . We can fold out the square terms from (5), producing

$$\begin{aligned} E^2 &= \sum_{i=1}^n (u^T H_{x_i} H_{x_i}^T u - 2y_i H_{x_i}^T u + y_i^2) + v(u) \\ &= (u^T D u + f^T u + \sum_{i=1}^n y_i^2) + v(u), \end{aligned} \quad (7)$$

where D and f are

$$D = \sum_{i=1}^n H_{x_i} H_{x_i}^T, \quad f = - \sum_{i=1}^n 2y_i H_{x_i}$$

Equation (7) is a quadratic function and thus we can add the constraints on u such that it represents a FM, producing a constraint QP. We can add the monotonicity constraint on u according to the properties P1 and P2 as $Cu \leq 0$, such that

$$C = \begin{pmatrix} M_1^T \\ M_2^T \\ \cdot \\ \cdot \\ M_{n+1}^T \\ \cdot \\ \cdot \\ M_{m(2^{m-1}-1)}^T \end{pmatrix}, \quad (8)$$

where $M_1^T \dots M_{m(2^{m-1}-1)}^T$ are vectors representing monotonicity constraint such as the one used in this work i.e. $g\{x_1\} - g\{x_1 \cup x_2\} \leq 0$ (see [5] for more details on C). Thus, the full QP to learn FM u is

$$\min_u 0.5u^T \hat{D}u + f^T u + v(u), \quad Cu \leq 0, \quad (0, 1)^T \leq u \leq 1, \quad (9)$$

where $\hat{D} = 2D$. We test the performance using ℓ_1 regularisation, i.e.

$$\min_u 0.5u^T \hat{D}u + f^T u + \lambda \|u\|_1, \quad (10)$$

where λ is the regularisation weight. The QPs at (9) and (10) provide a method to learn the FM u (i.e. g) from the training data. A new feature vector x' , from a test set, can thus be classified using the following steps:

1. Compute the normalised SVM decision value $f_k(x')$,
2. Apply the CFI at equation 1 with respect to the learned FM g ,
3. Compute the class label using $sign(f_k(x'))$.

2.5 State-of-the-art ensemble methods

Adaboost: Adaboost was introduced by Freund et al. [24] in 1997 which uses training sets to serially train each classifier and accords higher weight to the instances which are difficult to classify, with the objective of correctly classifying these in the next iteration [25]. Hence, after each iteration the weights of the misclassified instances are increased (which was initially equal for all instances) and the weights of the correctly classified instances are decreased. Moreover depending upon the overall accuracy, an additional weight (higher weight is assigned to more accurate classifiers) is assigned to each individual classifier, which is further used in the test phase. The sum of the weighted predictions is the final output of the ensemble model. The experimental settings and the base algorithm for the Adaboost are further discussed in the section 3.2

Bagging: Bagging (Bootstrap Aggregation) is an ensemble method introduced by Breiman et al. in 1996 [26], which aims to increase accuracy by combining the outputs of the classifiers in the ensemble. Sampling with replacement is used to train all the classifiers in the ensemble and thus some of the instances may appear more than once in the training set. Each classifier returns the class predictions for the test instances, and combines them using majority voting over all the class labels. Bagging is effective on unstable learning algorithms such as neural networks and decision trees [27] and thus we have chosen decision trees as the base classifier, discussed later in section 3.2.

Majority Voting with SVM (MJSVM): Let x be an instance and S_i (where $i = 1, 2, \dots, k$) be a set of base classifiers (Support Vector Machines) that output class labels $m_i(x, c_j)$ for each class label c_j (where $j = 1 \dots n$). The output of the final classifier $y(x)$ for instance x is given by

$$y(x) = \max_{c_j} \sum_{i=1}^k m_i(x, c_j). \quad (11)$$

More details on the MJSVM are described in the Experimental settings section 3.2.

Random Forest: A random forest is a collection of randomised decision trees where each decision tree is learned from different subsets of samples. The random forest classifier, in particular, needs two parameters: the number of classification trees (k), and the number of prediction variables to grow the trees (m) [28]. To

Table 2: Comparison of ensemble classification methods

Dataset Name	Binary Classes	No. of Features	No. of Instances
Dermatology	{1,2,3} vs {5,6,7}	33	366
Wine	{1} vs {2,3}	13	178
Ecoli	{1,2,5,8} vs {3,4,6,7}	7	336
Glass	{1,2,3} vs {5,6,7}	9	214
Sonar	{1} vs {2}	60	208
Ionosphere	{0} vs {1}	34	351
SPECTF Heart	{0} vs {1}	44	267
Bupa	{1} vs {2}	6	345
WDBC	{M} vs {B}	30	569
Haberman	{+} vs {-}	3	306
Pima	{+} vs {-}	8	768
Australian	{0} vs {1}	14	690
SA Heart	{0} vs {1}	9	462
Satimage	{1,2,3} vs {4,5,6,7}	36	6,435
Segmentation	{1,2,3,4} vs {5,6,7}	19	2,310
Mammographic	{0} vs {1}	5	830
Credit-approval	{+} vs {-}	15	653
Ozone	{0} vs {1}	72	1,848
Tic-tac-toe	{+} vs {-}	9	958
ltpd	{1} vs {2}	7	583

classify a test sample each tree is traversed and a vote is assigned to the class based on the probability score. The output is selected by choosing the mode i.e. the output with most votes, of all the ' k ' classification outputs. Reducing the number of predictive variables ' m ' reduces the correlation between trees, which stops ensemble model from converging to similar generalisation error and in turn helps in increasing the accuracy. Thus, ' m ' needs to be optimised to minimise the generalisation error.

3 Materials and Methods

3.1 Datasets and Pre-processing

20 benchmark datasets from the UCI machine learning repository [29] were selected to compare the performance of the selected algorithms, as shown in Table 2. These selected datasets contain different range of number of instances with different type of datasets. Not all the UCI datasets are binary and thus in some cases multiple classes are joined together for the purpose of binary classification [30].

The next step is data pre-processing to standardise the datasets for an unbiased comparison. All the data which had missing values were deleted and made homogeneous, i.e. all numeric. This might also help to locate inconsistencies among the data. Each dataset was processed using z-score [31] normalisation i.e. zero mean and unit standard-deviation. No further processing techniques were used as: 1) the aim of this work is not to report the best possible result for each

dataset, but to compare the performance of the classifiers. 2) to improve the classification results, further processing specific to each dataset would be required, leading to more challenging comparison [30].

3.2 Experiments

The results were produced by running each dataset for 100 trials. In each trial, 80% of the data were randomly used for training the classifiers and the remaining 20% for testing. Subsequently, the accuracies were statistically compared using a two-sample t-test.

The Adaboost ensemble was run with 200 decision trees, Bagging also with 200 but Random Forest ensemble method with 100 decision trees. The DeFIMKL and the MJSVM ensemble methods used the Support Vector Machine (SVM) algorithms with Radial Basis Function (RBF) kernels as their base classifiers. Five RBF kernels with their width (σ) evenly spaced between $0.5 - 1.5/(\text{number of features})$ were used for both the ensemble methods. Additionally, $L1$ regularisation with $\lambda = 0.5$ was used for all the datasets. The focus of this work is to show the effect on the final output with change in the aggregation models (DeFIMKL and MJSVM), and thus the settings for the RBF kernels and other ensemble methods were kept the same for all the datasets.

An underlying issue with ensemble classification algorithms is determining the size of ensemble. Not much discussion is given to this parameter selection as they are not in the scope of the paper.

4 Results and Discussions

Table 3 reports the average accuracies of the DeFIMKL, MJSVM, Adaboost, Bagging and Random Forest ensemble classification algorithms with standard deviations over 100 runs. A series of two-sample t-tests were conducted, which compared the accuracy of each algorithm against that of the highest performing algorithm for each dataset. To illustrate the results of these tests, both the absolute highest performing algorithm, along with any further algorithms that were found not to have a significantly lower (at $p < .05$) accuracy, are highlighted in bold. Thus, the values highlighted in bold should represent the algorithm(s) with the highest accuracy for each dataset, and will include all values that were not found to significantly differ from the best-performing model.

The results for the comparison of FM-FI based ensemble classification algorithm DeFIMKL show promising results on the selected UCI datasets. It can be observed that DeFIMKL has good performance in 16 out of 20, whereas MJSVM, Adaboost, Bagging and Random Forest were best in 15, 6, 10 and 7 datasets respectively. DeFIMKL can more closely be compared with MJSVM ensemble classification algorithm as the difference between the two algorithms is in the decision level fusion. MJSVM was one of the best algorithm in 15 out of the 20 selected datasets, one less than DeFIMKL. However, MJSVM has an advantage since it simply takes a majority vote rather than learning a (typically) very large FM whose length scales exponentially with the number of classifiers. It can

Table 3: Accuracy comparison of ensemble classification methods for the benchmark datasets*

Datasets	DeFIMKL	MJSVM	Adaboost with trees	Bagging	Random Forest
Dermatology	97.35 (1.86)	97.47 (1.75)	96.69 (2.05)	97.31 (1.77)	95.47 (2.65)
Wine	99.44 (1.18)	99.42 (1.44)	96.81 (2.91)	97.78 (2.68)	96.44 (3.14)
Ecoli	96.77 (1.84)	96.84 (1.85)	95.57 (2.01)	96.34 (2.36)	95.82 (2.29)
Glass	94 (3.8)	94.16 (3.73)	93.91 (3.92)	94.28 (3.13)	92.72 (3.78)
Sonar	84.57 (4.7)	83.76 (5.4)	83.17 (5.89)	83.21 (6.2)	79.31 (6.2)
Ionosphere	94.61 (2.71)	94.34 (2.59)	90.41 (3.72)	92.76 (2.74)	91.15 (3.57)
SPECTF Heart	79.19 (4.4)	79.48 (4.18)	79.78 (4.98)	81.85 (5.01)	80.81 (4.23)
Bupa	69.88 (4.95)	69.77 (4.71)	70.93 (5.32)	70.43 (5.02)	69.77 (5.65)
WDBC	97.22 (1.64)	97.28 (1.62)	96.59 (2.03)	95.51 (1.94)	95.11 (2.15)
Haberman	73.77 (4.84)	73.92 (4.88)	73.4 (4.25)	69.34 (4.99)	69.24 (4.52)
Pima	76.12 (3.17)	76.49 (3.14)	75.97 (3.22)	76.61 (3.16)	75.97 (3.21)
Australian	85.75 (2.42)	85.85 (2.35)	85.94 (2.79)	86.96 (2.68)	86.37 (2.81)
SA Heart	71.16 (4.04)	71.66 (3.89)	69.31 (4.23)	69.39 (4.23)	68.63 (4.07)
Satimage	95.69 (0.58)	95.53 (0.56)	94.09 (0.6)	95.6 (0.47)	95.17 (0.69)
Segmentation	91.81 (1.15)	91.46 (1.08)	92.86 (1.53)	96.03 (1.1)	96.29 (1.26)
Mammographic	82.16 (3.65)	82.88 (4.02)	81.8 (5.71)	78.6 (5.39)	77.21 (5.02)
Credit-approval	86.89 (2.38)	86.66 (2.73)	86.47 (2.84)	87.4 (3.15)	87.02 (2.83)
Ozone	97.12 (0.79)	97.1 (0.8)	96.76 (0.77)	96.85 (0.85)	96.64 (0.83)
Tic-tac-toe	89.64 (2.25)	88.54 (2.39)	84.34 (2.5)	94.38 (1.85)	95.35 (1.95)
lpld	72.18 (3.58)	71.79 (3.58)	68.18 (4.49)	68.01 (4.38)	67.77 (4.15)

*The numbers report the average accuracies with standard deviations over 100 runs

thus be inferred that the non-linear CFI aggregation can squeeze out a better performance at the cost of additional memory complexity, although a detailed comparison is need to fully answer this question.

5 Conclusions and Future Works

FIs are aggregation operators which are capable of exploiting all the possible interactions among the inputs through the use of FMs. Thus, in the context of ensemble classification FI-FM based ensemble classifiers not only consider individual classifier in the ensemble but all the possible combinations of classifiers in the ensemble. FI-FM ensemble classifiers have been used in various applications over the past decade, yet in most papers, a dataset oriented comparison with state-of-the-art ensemble classification has been made. Thus, in this paper we addressed this gap and compared a FI-FM based ensemble classification algorithm DeFIMKL with the state-of-the-art ensemble classification algorithms Adaboost, Bagging, MJSVM and Random Forest over 20 datasets.

DeFIMKL was one of the best algorithms among 16 out of 20 selected datasets and fell very close to the best in the remaining datasets, with the exception of the Segmentation and tic-tac-toe dataset. On comparing the accuracies, it can be observed that DeFIMKL is either equivalent or better in performance than MJSVM. However, MJSVM has an advantage as it reduces the memory

complexity without having to learn the huge 2^n FM vector i.e. taking the majority voting of the SVMs results in similar performance to the non-linear CFI aggregation of the SVMs.

It is important to note that the DeFIMKL and the MJSVM are only compared for one base classifier i.e. SVM. Thus, in future, we aim to include different base classifiers such as Neural Networks, Decision Trees, etc, to fully answer the question raised in the paper. The results suggest that DeFIMKL achieves good accuracy over a number of datasets, although a time complexity comparison is also needed to answer the effect of base classifiers on the FI-FM ensemble methods. Thus, in future we also aim to compare the time complexity among the selected ensemble classifiers.

Acknowledgements: The research was supported by The University of Nottingham Vice-Chancellor’s Scholarship for Research Excellence (International).

References

1. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: a survey. *Information Fusion* **37** (2017) 132–156
2. Domingos, P.: A few useful things to know about machine learning. *Communications of the ACM* **55**(10) (2012) 78
3. Islam, M.A., Anderson, D.T., Pinar, A.J., Havens, T.C.: Data-Driven Compression and Efficient Learning of the Choquet Integral. *IEEE Transactions on Fuzzy Systems* **PP**(99) (2017)
4. Pinar, A.J., Rice, J., Hu, L., Anderson, D.T., Havens, T.C.: Efficient Multiple Kernel Classification using Feature and Decision Level Fusion. In: *IEEE Transactions on Fuzzy Systems*. Volume 6706. (2016) 1–1
5. Anderson, D.T., Price, S.R., Havens, T.C.: Regularization-based learning of the Choquet integral. In: *IEEE International Conference on Fuzzy Systems*, IEEE (jul 2014) 2519–2526
6. Pinar, A.J., Havens, T.C., Islam, M.A., Anderson, D.T.: Visualization and learning of the Choquet integral with limited training data. *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (jul 2017) 1–6
7. Pinar, A.J., Anderson, D.T., Havens, T.C., Zare, A., Adeyeba, T.: Measures of the Shapley index for learning lower complexity fuzzy integrals. *Granular Computing* **2**(4) (dec 2017) 303–319
8. Pinar, A., Havens, T.C., Anderson, D.T., Hu, L.: Feature and decision level fusion using multiple kernel learning and fuzzy integrals. *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (aug 2015) 1–7
9. Lequn Hu, Anderson, D.T., Havens, T.C.: Multiple kernel aggregation using fuzzy integrals. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (2013)
10. Li, K., Chen, C., Liu, W., Fang, X., Lu, Q.: Software defect prediction using fuzzy integral fusion based on GA-FM. *Wuhan University Journal of Natural Sciences* **19**(5) (2014) 405–408
11. Zhang, L., Zhou, D.Q., Zhou, P., Chen, Q.T.: Modelling policy decision of sustainable energy strategies for Nanjing city: A fuzzy integral approach. *Renewable Energy* **62** (2014) 197–203
12. Cavrini, F., Bianchi, L., Quitadamo, L.R., Saggio, G.: A Fuzzy Integral Ensemble Method in Visual P300 Brain-Computer Interface. *Computational Intelligence and Neuroscience* **2016** (dec 2016) 1–9

13. Karczmarek, P., Pedrycz, W., Reformat, M., Akhouni, E.: A study in facial regions saliency: A fuzzy measure approach. *Soft Computing* **18** (2014) 379–391
14. Wang, Z., Xiao, N.: Fuzzy Integral-based Neural Network Ensemble for Facial Expression Recognition. *Proceedings of the International Conference on Computer Information Systems and Industrial Applications* (2015)
15. Anderson, D.T., Havens, T.C., Wagner, C., Keller, J.M., Anderson, M.F., Wescott, D.J.: Extension of the fuzzy integral for general fuzzy set-valued information. *IEEE Transactions on Fuzzy Systems* **22**(6) (dec 2014) 1625–1639
16. Pinar, A.J., Rice, J., Havens, T.C., Masarik, M., Burns, J., Anderson, D.T.: Explosive hazard detection with feature and decision level fusion, multiple kernel learning, and fuzzy integrals. *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016* (dec 2017) 1–8
17. Wagner, C., Havens, T.C., Anderson, D.T.: The Arithmetic Recursive Average as an Instance of the Recursive Weighted Power Mean. In: *IEEE Int. Conf. Fuzzy Systems*. 2017. (2017)
18. Fakhar, K., El Aroussi, M., Saidi, M.N., Aboutajdine, D.: Applying the upper integral to the biometric score fusion problem in the identification model. *International Conference on Electrical and Information Technologies (ICEIT)* (2015)
19. Wang, Q., Zheng, C., Yu, H., Deng, D.: Integration of heterogeneous classifiers based on choquet fuzzy integral. *7th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2015* **1** (2015) 543–547
20. Murofushi, T., Sugeno, M.: A learning model using fuzzy measures and the Choquet integral. *Proc. of the 5th Fuzzy System Symposium* **29** (1989) 213–218
21. Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research* **89**(3) (1996) 445–456
22. Ko, Y.C., Fujita, H., Tzeng, G.H.: An extended fuzzy measure on competitiveness correlation based on WCY 2011. *Knowledge-Based Systems* **37** (2013) 86–93
23. Anderson, D.T., Keller, J.M., Havens, T.C.: Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral. *Lecture Notes in Computer Science* **6178 LNAI** (2010) 502–511
24. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *journal of computer and system sciences* **55** (1997) 119–139
25. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* **42**(4) (jul 2012) 463–484
26. Breiman, L.: Bagging predictors. *Machine Learning* **24**(2) (1996) 123–140
27. Sun, Q., Pfahringer, B.: Bagging Ensemble Selection. *Proceedings of the 24th international conference on Advances in Artificial Intelligence* (2011) 251–260
28. Rodriguez-Galiano, V.F., Ghimire, B., Rogan, J., Chica-Olmo, M., Rigol-Sanchez, J.P.: An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing* **67**(1) (2012) 93–104
29. Lichman, M.: {UCI} Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences (2013)
30. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D., Amorim Fernández-Delgado, D.: Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research* **15** (2014) 3133–3181
31. Patro, S.G.K., Sahu, K.K.: Normalization: A Preprocessing Stage. *arXiv preprint arXiv:1503.06462*. (2015) 4