Supervised Radar Signal Classification

Ivan Jordanov
School of Computing,
University of Portsmouth,
Portsmouth, UK
Ivan.Jordanov@port.ac.uk

Nedyalko Petrov
GSTT/KCL Comprehensive
Biomedical Research Centre,
King's College, London, UK
Nedyalko.Petrov@kcl.ac.uk

Alessio Petrozziello
School of Computing,
University of Portsmouth,
Portsmouth, UK
Alessio.Petrozziello@port.ac.uk

***Abstract* - This work investigates radar signal classification and source identification using three classification models: Neural Networks (NN), Support Vector Machines (SVM) and Random Forests (RF). The available large dataset consists of pulse train characteristics such as signal frequencies, type of modulation, pulse repetition intervals, scanning type, scan period, etc., represented as a mixture of continuous, discrete and categorical data. Typically, considerable part of the data samples contains missing values. In our previous work we used only part of the radar dataset, applying listwise deletion to clean the samples with missing values and processed relatively small subset of complete data. In this work we apply three different imputation techniques to deal with the missing data: multiple imputation (MI), K-Nearest Neighbour Imputation (KNNI) and Bagged Tree Imputation (BTI). We employ the imputation methods to all data samples with up to 60% missingness, this way increasing more than twice the size of the initially used data subset. Subsequently the three classifiers (NN, SVM, and RF) are employed and the results are analysed and critically compared based on their accuracy to assess the model with the best performance.**

Keywords—supervised classification with NN, RF and SVM; missing data; multiple imputation; KNN imputation; BT imputation; supervised learning; radar signal classification

I. Introduction

Established machine learning models as Neural Networks (NN), Support Vector Machines (SVM) and Random Forests (RF) are considered appropriate and efficient approaches for classifying labelled data (supervised learning). Nevertheless, before employing them for solving particular problems, the datasets at hand usually need pre-processing stage, which includes dealing with issues related to the data quality – balanced and unbalanced data, presence of noise and outliers, and especially missing data. Mechanisms of missing data are generally considered to belong to three large categories [1]: missing at random (MAR), where the missingness may depend on observed data but not on unobserved data (in other words, the cause of missingness is considered); missing completely at random (MCAR) – a special case of MAR, where the probability that an observation is missing is unrelated to its value or to the value of any other variable; and missing not at random (MNAR), where the missingness depends on unobserved data. The last group (MNAR) usually yields biased parameter estimates, while MCAR and MAR analyses yield unbiased ones (at the same time the main MCAR consequence is a loss of statistical power).

Ideally, dealing with missingness requires an analysis strategy that leads to least biased estimates, and at same time without losing statistical power. The problem is that these are contradictory criteria and using the information from the partial data in missing data samples (keeping the statistical power), while substituting the missing data samples with estimates, inevitably brings biases.

There are two groups of methods widely used in dealing with missingness: Deletion methods and Model-based methods [1]. In this paper we consider three model-based approaches: Multiple Imputation (MI), Bagged Tree Imputation (BTI) and K-Nearest Neighbour Imputation (KNNI).

The MI approach involves three distinct steps: firstly, sets of plausible data for the missing observations are created and these sets are filled in separately to create many 'completed' datasets; secondly, each of these datasets is analysed using standard procedures for complete datasets; and thirdly, the results from the previous step are combined and pooled into one estimate for the inference. The aim of the MI process is to fill in the missing values with plausible estimates, plugin in multiple times these values, while preserving important statistical characteristics of the whole dataset. As with most (multiple regression) prediction models, there is a danger of overfitting the data, which could lead to less generalizable results than the original data would have [2].

The KNNI [3] approach initially will take only rows of the dataset without missing data and will use these as a prototype dataset to select the nearest one; subsequently will choose a distance metric and compute the nearest neighbour between each pattern with missing data and the full dataset; and finally, will impute the data, using the mean or the mode of the chosen neighbours.

The BTI is a machine learning technique used for solving regression problems, which produces prediction model in the form of an ensemble of weak prediction models. For each predictor that has missing data, a tree model is trained, based on the other predictors and the values are imputed using a regression function [4, 5].

II. Supervised Radar signal classification

The radar characteristics (range, resolution, sensitivity, etc.) are determined by its transmitter and waveform generator. Most of the radars operate within the microwave region of the electromagnetic spectrum with frequency range of about 200 MHz to 95 GHz, and are

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

used for short range application with high resolution, while other radars operate at a very low frequency bands and are usually preferred for longer range purposes [6].

Intercepted and collected pulse train characteristics typically include signal frequencies, type of modulation, pulse repetition intervals, etc. The collected information usually consists of a mixture of continuous, discrete and categorical data, and also frequently includes missing values. Missing values are imminent part of real world datasets and radar datasets make no exception. There is a variety of reasons why data may be missing, but common ones are related to human factor, equipment malfunction or coarse environment conditions that result in noise and propagation distortion, leading to incomplete, erroneous or missed to intercept signals [5]. Table I shows several samples of radar data that consist of continuous, discrete and categorical data, as well as include missing values.

A. Neural Networks

Various approaches and methods have been used for radar emitter recognition and classification, and considerable part of this research incorporates Neural Networks (NN), because of their massively parallel architecture, fault tolerance and ability to handle incomplete radar type descriptions. NN techniques have previously been applied to several aspects of radar localisation, analysis and identification of radiated electromagnetic energy processing [7]. More recently, many new radar recognition systems have been using neural networks as a key classifier. Examples of a variety of NN architectures and topologies used for radar identification, recognition and classification include popular multilayer perceptron (MLP), radial basis function networks (RBF), vector neural networks, single parameter dynamic search neural networks, and others [8-11]. Deep architecture NN have also been used for classification of synthetic aperture radar images [12].

B. Support Vector Machines

Support Vector Machines (SVM) are paradigms based on statistical learning theory, that make use of polynomial classifiers, neural networks, and RBF networks as special cases [13, 14]. They minimize the structure risk in high dimensional feature space searching an optimal discriminant hyperplane with a large margin between the classes. If the feature space is linearly non-separable, the SVM use non-linear mapping to find an optimal classification hyperplane. The choice of a map function (kernel) is of critical importance and can substantially determine the classification results [14]. One advantage of this approach is the possibility to design and use a kernel for a specific problem that could be applied to the data without the need of a feature extraction process. The SVM have been used recently to classify radar pulse signals. For example, in [13, 15] the authors implement SVM to classify targets by using a micro-Doppler features when analysing micro-motions of an object having single signature. The reported results show low error rate, demonstrating the ability of SVM to tackle and solve similar problems successfully. Implementations of SVM exist in almost every programming language and at least four R packages contain SVM related software (we used the R package [16], which we found to have better interface and to work faster on multi-class problems).

C. Random Forests

Random Forests (RF) [17] is a powerful machine learning technique that operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Each tree is trained on a bootstrapped sample of the training data, and at each node, the algorithm only searches across a random subset of the variables to determine a split. To classify an input vector, it is submitted to each of the decision trees in the forest and the prediction is then formed using a majority vote. The method combines Breiman's "bagging" idea and the random selection of features, introduced independently by Amit and Geman [17].

Key advantages of RF include: their non-parametric nature; avoiding decision trees' habit of overfitting; high classification accuracy; and capability to determine variable importance [18]. However, as the split rules for classification are unknown, the RF can be considered to be a black box type classifier. In our implementation, we used "*randomForest*" function (from the R package), which implements the Breiman's algorithm [17, 19].

III. Problem Statement and Data Set Analysis

Reliable and real-time identification of radar signals is of crucial importance for timely threat detection, threat avoidance, general situation awareness and timely deployment of counter-measures. In this context, this paper investigates the potential application of a NN-based approach for timely and trustworthy identification of radar types, associated with intercepted pulse trains.

In our previous work [20], we initially used listwise deletion to obtain 7693 samples with no missing data from a total of 29094 intercepted generic data samples and subsequently implemented multiple imputation (MI), to make use of the whole dataset when solving identification and classification problem. Each of the captured signals in the dataset is pre-classified by experts into one of 125 categories, based on the main radar emitter functions (e.g., surveillance, air defence, air traffic control, weather tracking, etc.).

In this work we investigate three different approaches for the classification. In the first one we employ MI and use 15656 samples and recovered 7963 samples with missing data (doubled the number of the available data). When employing KNNI and BTI we managed to recover the whole missing data and used 29094 samples (the whole dataset). From the missing data samples (example given in Table I), we excluded only those with above 60% missingness and used the imputation methods for substituting the missing data. In Table I, the first column is a data sample identifier, the second is a category label (specifying the radar function and considered as a system output) and the rest are radar signal pulse train characteristics (e.g., signal frequencies, type of modulation, pulse repetition intervals, etc.), which are considered as input parameters.

A more comprehensive summary of the data is presented in Table II, where an overview of the type, range and percentage of missing values for the parameters in the data set is given. The included data consists of both numerical (integer and float) and categorical values,

which we coded during the data pre-processing stage, in order to convert them into numerical representations. The table also shows that the percentage of missing data varies from 11.2% for the *RF* variable to 59.4% for the *SP* variable.

TABLE I. SAMPLE RADAR DATA SUBSET. MISSING VALUES (VALUES THAT COULD NOT HAVE BEEN INTERCEPTED OR RECOGNIZED) ARE DENOTED WITH '-'. THE REST OF THE ACRONYMS ARE DEFINED IN TABLE II

ID	FN	RFC	RF mi	RF ma	PRC	PRI mi	PRI ma	PDC	PD mi	PD ma	ST	SP mi	SP ma
84	SS	B	5300	5800	K	–	–	S	–	–	A	5.9	6.1
4354	AT	F	2700	2900	F	1351.3	1428.6	S	–	–	A	9.5	10.5
7488	3D	B	8800	9300	K	100	125	S	13	21	B	1.4	1.6
9632	WT	F	137	139	T	–	–	V	–	–	D	–	–
9839	3D	S	2900	3100	J	–	–	V	99	101	A	9.5	10.5

TABLE II. DATA DESCRIPTION AND PERCENTAGE OF MISSING VALUES. IN COLUMN 'TYPE': I – INTEGER; C – CATEGORICAL; R – REAL VALUES

Field	Field Description	Type	Levels	% Missing
ID	Reference for the line of data	I	-	-
FN	Function performed by the radar ('*3D'* – 3D surveillance, '*AT'* – air traffic control, '*SS'* – surface search, '*WT'* – weather tracker, etc.)	C	142	1.4
RFC	Type of modulation used by the radar to change the frequency from pulse to pulse ('*A'* – agile, *'B'* – burst transmissions, '*S*' – scan, '*F'* – fixed, etc.)	C	12	20.7
RFmi	Min frequency used by the radar	R	-	11.2
RFma	Max frequency used by the radar	R	-	11.2
PRC	Type of modulation used by the radar to change the Pulse Repetition Interval (PRI), ('*F'* – fixed, '*J*' – jitter, '*K*' – constant stagger, '*T*' – tone, etc.)	C	15	15
PRImi	Min PRI used by the radar	R	-	46.7
PRIma	Max PRI used by the radar	R	-	46.7
PDC	Type of modulation used by the radar to change the pulse duration ('*S'* - stable)	C	5	12.9
PDmi	Min pulse duration used by the radar	R	-	46.1
PDma	Max pulse duration used by the radar	R	-	46.1
ST	Scanning type – used method by the radar to move the antenna beam ('*A'* – circular, '*B'* – bidirectional, '*W'* – electronically scanned, etc.)	C	28	11.3
SPmi	Min scan period used by the radar	R	-	59.4
SPma	Max scan period used by the radar	R	-	59.4

IV. DATA IMPUTATION AND PRE-PROCESSING

The pre-processing of the available data is of a great importance for the subsequent machine learning stage as it can affect significantly the overall success or failure of the employed classification algorithm. In this context, the main objective at this stage is to analyse the data for inconsistences, outliers and irrelevant entries, and to transform it into a form that could facilitate the underlying mathematical apparatus of the machine learning method that can lead to an overall improvement of the classifier's performance.

A. Data Imputation

1) Bagged Tree Imputation (BTI)

Bagging predictors is a method of generating multiple versions of a predictor in order to get an aggregated one. The aggregation uses the average over the versions of the predictor when we want to predict a numerical outcome, and employs a plurality vote when the predicting outcome is a class. The multiple versions are formed by making bootstrap replicates of the training set and subsequently using these as new learning sets. Tests on real and simulated data sets, using classification and regression trees and subset selection with linear regression, show that bagging can benefit the accuracy. The vital element in this approach is the instability of the prediction model, but if perturbing the learning set can cause significant changes in the constructed predictor, then bagging can improve the accuracy [4, 5].

Bagging proved to be more efficient in the presence of label noise when compared to boosting an d randomisation [21], it is also robust to outliers and is able to impute the data very accurately using surrogate splits [22]. The latter feature (surrogate splits) is essential in handling missing data. For instance, say a decision tree is trained to predict variable *d*, using variables *a*, *b* and *c*, and if we receive values only of *a* and *b*, the missing value of *c* would raise problems regarding the prediction of *d*. In such cases, the tree models are trained to include surrogate splits. So, if the variable *c* is missing in a new data point, the algorithm defers the decision to another variable that is highly correlated to the missing variable *c*, which will allow the prediction to continue. Another important feature of the tree model is its flexibility, which makes possible to train different models with the RF and defer the real prediction to a system vote among them.

In this work we employ gradient boosting technique for the values regression, which uses an ensemble of weak decision trees. For this purpose, we used the R "*gbm*" package [19] to create our imputation function.

2) K- Nearest Neighbour Imputation (KNNI)

Another imputation method used in this work is the K-Nearest Neighbour (KNNI) [3] method. This approach has a number of benefits: the method can predict both, categorical variables (the most frequent value among the KNN) and continuous variables (the average among the KNN); and when using this imputation, there is no need to build a model (as in the Bagged Tree Imputation).

If the available data set is denoted with Y and the complete subset with Y_c, for each sample containing a missing point, the procedure will search the k most similar samples in the Y_c dataset, according to adopted distance measure, before imputing the missing value. This means that there is a need to separate the complete dataset from the whole dataset containing missing data, define a suitable distance measure, and for each sample in the missing dataset to look for the KNN, using that distance

measure. Then, the substitution uses the most frequent value for a categorical variable and the mean value for a continuous variable. In this work we used the function *kNN* of the ***R*** package VIM [23, 24].

3) Multiple Imputation

For imputing the missing multivariate data we use the sequential imputation algorithm [25], implemented in the *impSeq* function from the ***R*** package (we also tried two other ***R*** functions: *impNorm;* and *impSeqRob*; but they didn't produce better results when tested on the complete dataset).

If Y_c is the complete subset from the available data set Y, the procedure will start with Y_c to estimate sequentially the missing values of an incomplete observation y^*, by minimizing the covariance of the augmented data matrix $Y^* = [Y_c, y^*]$. Subsequently the data sample y^* is added to the complete data subset and the algorithm continues with the estimate of the next data sample with missing values.

Because *impSeq* uses the sample mean and covariance matrix, it is vulnerable to the presence of outliers, but this can be amended by including robust estimators of location and scatter (which is realised in the *impSeqRob* function) [25]. Since the outlyingness metric can only be computed for a complete dataset, the sequential imputation of the missing data is done first and then the outlyingness measure is computed and used to define whether the observation is an outlier or not.

As we mentioned before, in the available radar dataset of 29094 samples, there are 7693 fully intercepted and recognised radar signals that constitute the complete subset (received after listwise deletion of the original dataset) [20]. Subsequently, employing the MI on the missing data samples with less than 60% missingness, led to dataset of 15656 observations, which is more than the doubled size of the initial data subset. Processing the missing subset with the KNNI and the BTI tripled the complete data samples, enabling us to utilise valuable information and use the statistical power of the data contained in the samples with missing values.

TABLE III. SAMPLE RADAR DATA SUBSET WITH IMPUTED VALUES FOR THE MISSING CONTINUOUS VALUES

ID	FN	RFC	RF mi	RF ma	PRC	PRI mi	PRI ma	PDC	PD mi	PD ma	ST	SP mi	SP ma
84	SS	B	5300	5800	K	963.2	5625	S	5.8	17	A	5.9	6.1
4354	AT	F	2700	2900	F	1351	1428	S	4	6.3	A	9.5	10.5
7488	3D	B	8800	9300	K	100	125	S	13	21	B	1.4	1.6
9632	WT	F	137	139	T	622.6	31312	V	61.1	93.1	D	12	47.8
9839	3D	S	2900	3100	J	2058	48128	V	99	101	A	9.5	10.5

For the identification and classification of the radar signals, the applied supervised learning uses from two to eleven output classes: in the first set of simulations we use 2 classes – civil and military (defined by experts in the field from a total of 125 functional categories); and in the second set of simulations - four civil and seven military classes, which gives eleven output labels to classify. Table III shows the samples from Table I with the inputted values produced by the implemented MI.

B. Data Coding and Transformation

This stage of the pre-processing aims to transform the data into a form suitable for feeding to the selected classifier, expecting to facilitate faster and more accurate machine learning. In particular, a transformation coding is applied to convert the categorical values presented in the data set to numerical ones. Three of the most broadly applied coding techniques are investigated and evaluated here: continuous; binary; and dummy variables.

TABLE IV. SAMPLE SUBSET WITH IMPUTED RADAR DATA AND NATURAL NUMBER CODING OF 'RFC', 'PRC', 'PDC', AND 'ST'

ID	RFC	RF mi	RF ma	PRC	PRI mi	PRI ma	PDC	PD mi	PD ma	ST	SP mi	SP ma
84	2	5300	5800	7	963.2	5625	1	5.8	17	1	5.9	6.1
4354	4	2700	2900	4	1351	1428	1	4	6.3	1	9.5	10.5
7488	2	8800	9300	7	100	125	1	13	21	2	1.4	1.6
9632	4	137	139	11	622.6	31312	2	61.1	93.1	4	12	47.8
9839	9	2900	3100	6	2058	48128	2	99	101	1	9.5	10.5

For the first type of coding, each of the categorical values is substituted by a natural number, e.g., the 12 categories for the RFC input are encoded with 12 ordinal numbers, the 15 PRC categories – with 15 ordinal numbers, etc. A sample of a data subset coded with continuous values is given in Table IV. Binary coding, wherein each non-numerical value is substituted by $\log_2 N$ new binary variables (where N is the number of categories taken by that variable), each having value of either 0 or 1, is illustrated in Table V for 32 categories.

TABLE V. EXAMPLE OF BINARY CODING FOR 32-LEVEL CATEGORICAL VARIABLE.

Original Category		Encoded Variables				
Index	Label	B1	B2	B3	B4	B5
1	'2D'	0	0	0	0	0
2	'3D'	0	0	0	0	1
3	'AA'	0	0	0	1	0
...						
16	'CS'	0	1	1	1	1
...						
32	'ME'	1	1	1	1	1

Finally, the non-numerical attributes are coded using dummy variables. In particular, every p levels of a categorical variable are represented by introducing p dummy variables. An example of dummy coding for 32 categorical levels is shown in Table VI.

TABLE VI. EXAMPLE OF DUMMY CODING FOR 32-LEVEL CATEGORICAL VARIABLE

Original Category		Encoded Variables								
Index	Label	D1	D2	D3	D4	D5	···	D16	···	D32
1	'2D'	1	0	0	0	0	···	0	···	0
2	'3D'	0	1	0	0	0	···	0	···	0
3	'AA'	0	0	1	0	0	···	0	···	0
...										
16	'CS'	0	0	0	0	0	···	1	···	0
...										
32	'ME'	0	0	0	0	0	···	0	···	1

Taking into account the large number of categories present for the categorical attributes in the input data set (Table II), continuous and binary coding were used for transforming the input variables. On the other hand, binary and dummy variable coding were chosen for representing the output parameters.

In order to balance the impact of the different input parameters on the training algorithm, data scaling was also used. Correspondingly, each of the conducted experiments in the next section is evaluated using 3 forms of the input data set: the original data (with no scaling); normalized data (i.e., scaling the attribute values within (0, 1) interval); and standardized data (i.e., scaling the attribute values to a zero mean and unit variance).

V. Imputation and Assessment

To begin with, it is essential to determine which imputation method leads to the best substitution of the missing values. For this purpose, we designed two experiments, as described below.

A. Label Imputation

In this case, in order to test the reliability of the investigated imputation models, 25% of the labels in the complete data subset (~4000 samples), were randomly removed. The labels were subsequently imputed, using the three methods: MI, BTI and KNNI. Considering the random nature of the algorithms, we run the imputation 30 times for each of them. This was done for the two simulations: with 2 classes and with 11 classes.

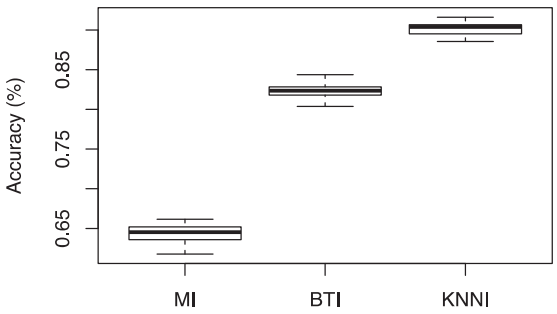

Fig. 1. Label imputation for MI, BTI and KNNI over 30 runs for 2 classes. Each boxplot displays the minimum and maximum values (whiskers), the first and the third quartile (boundaries of the box represent 50% of the data), and the median (the thick line).

In the 2 class imputation case (Fig. 1), the KNNI method showed the best performance with a 90% accuracy, followed by BTI with 84% and MI with 65%. We think the KNNI achieved the best accuracy due to the skewed nature of the data: in fact, 75% of the data belong to class 2 and in this way, the imbalance in the dataset influences the search of nearest neighbours to go towards the class with more samples. On the contrary, the MI and BTI methods assume balanced label distribution, which leads to limited accuracy of their imputation.

With 11 classes (Fig. 2), the effect is even more evident, the patterns are not equally distributed among the classes, providing a low accuracy for MI and BTI and a high variance for KNNI due to the randomness of the labels removed in each of the 30 runs.

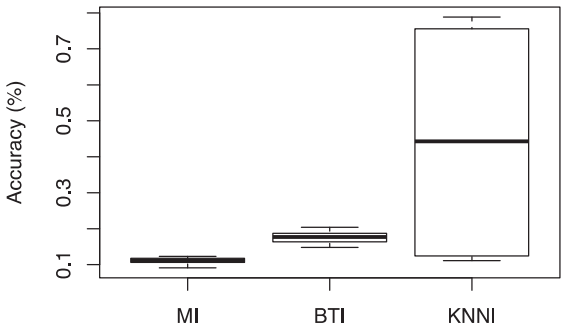

Fig. 2. Label imputation with MI, BTI and KNNI over 30 runs for the 11 classes. Each boxplot displays the minimum and maximum values (whiskers), the first and third quartile (boundaries of the box, represent 50% of the data) and the median (thick line).

B. Continuous features imputation

The second experiment aimed to validate the algorithms' accuracy in presence of continuous features. Eight features were tested from the dataset without missing data (*RFmi, RFma, PRImi, PRIma, PDmi, PDma, SPmi,* and *SPma*) (Table II). As in the previous experiment, 25% of the data were removed from each feature and subsequently imputed. The Root Mean Square Error (RMSE) between the imputed values and the real ones was selected to measure the methods' performance. The experiment was iterated 30 times and the non-parametric Wilcoxon test for statistical significance along with the relative effect size (Cohen's parameter *d*), were calculated [26].

The Wilcoxon test uses the rank of the data to determine if there is any difference between two samples, without making any assumption about the data distributions' nature. If the *p-value* is greater than the significance threshold α (α = 0.05 in our case), then there is no significant difference between the two samples.

The Cohen's *d* effect size measure shows how much, on average, one technique outperforms another. The measure applied to the two populations (1st technique vs the 2nd one) gives a response between 0 and 1 using the mean and the standard deviations of the two techniques [26].

The considered imputation algorithms were also compared with the median imputation (an algorithm from the "single imputation" family, largely rejected by the research community [2]) and for this reason, used just as a comparison baseline. As expected (Fig. 3), all other algorithms outperformed the median imputation, providing smaller errors.

In 22 out of 24 cases BTI was with better statistical significance (*p-value* < 0.025) than MI, KNNI and *median*, with *effect size d* = 1 (Fig.3 shows comparison of BTI with the three other methods for four of the variables). The comparison between BTI and KNNI on *PRImi* and *PRIma*, despite a *p-value* < 0.025, shows slightly lower *effect size*, respectively $d = 0.83$ and $d = 0.78$. Analysis of the distribution of those two features shows that KNNI is again strongly affected by the imbalance in the data (highly positively skewed for those two features).

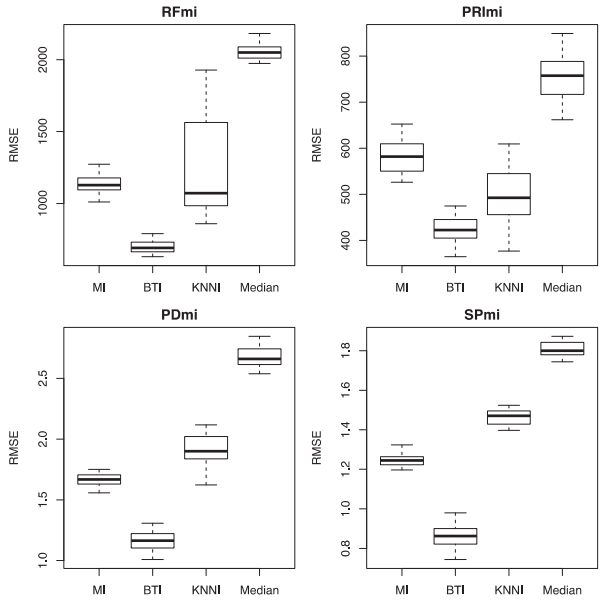

Fig. 3. Root mean square error (RMSE) for the imputation of the continuous values (30 runs). Low median values represent preferable imputation methods. Non-overlapped boxplots indicate statistical difference between the algorithms. For a better visualisation, the features RFma, PRIma, PDma and SPma have been omitted due to similar distribution and high correlation with the respective minima.

VI. Classifiers Training and Results

Two main experiments are conducted for investigating the efficiency of each classifier when solving the radar emitter classification problem.

Before the imputation, the data is split into two separate datasets: the first one for training (75% of the whole data) and the second one for testing (the remaining 25% of data).

The investigated neural network topologies include one hidden layer with fully connected neurons in the adjacent layers and batch-mode training. For the first experiment, the NN topology was *n-n-2*, where *n* is the number of inputs and the output contains 2 binary neurons coded as: 10 for class *"Civil"*; and 01 for class *"Military"*. For the second experiment, the same topology was used with 11 output neurons (representing 4 *civil* and 7 *military* classes). The training set was further divided into 80% for the training and 20% for the validation, and Mean Squared Error (MSE) was used for evaluating the learning performance. The stopping condition also included 500 training epochs, gradient reaching value less than 1.0e-06, or 6 consequent failed validation checks, whichever occurs first. NN learning with *Levenberg-Marquardt* algorithm was then used.

For the RF, the basic split (75% training – 25% testing) was used to perform the classification. The limit for the forests was set to 500 trees and the output class was decided by a vote among these.

The SVM were provided with a radial basis kernel (other kernels were tried as well but led to worse results) and the input parameters were generated randomly by the algorithm. Moreover, an optimisation function *tune.svm* [16], was used to tune the parameters of the model on the training set, which increased the accuracy of the classifier up to 4%.

The accuracy of each classifier (coding the categorical variables with the three techniques described above) was investigated, evaluated and compared, before and after the data normalisation and standardisation.

Sample confusion matrices are shown for the best accuracy achieved in the experiments for the 2 and 11 classes after training with continuous input data. The RF demonstrates high accuracy, as it can be seen from Fig. 4 and Fig. 5, showing the number of correct responses in the green squares and the number of incorrect responses in the red squares. The bottom right percentage illustrates the overall classifier accuracy (OCA).

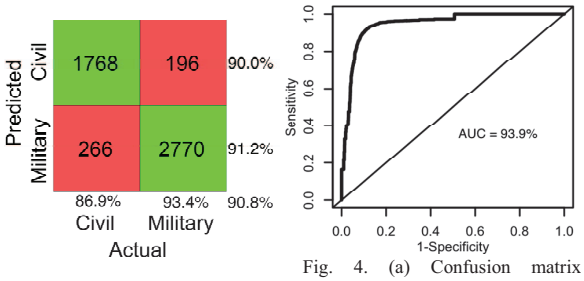

Fig. 4. (a) Confusion matrix illustrating the RF classification results for the two classes, after employing BTI with continuous value coding (*Military* and *Civil*). (b) ROC Curve for the same simulation.

Furthermore, it can be observed from Fig. 5 that the number of hits, as well as the accuracy of the RF classifier compared to our previous work [20], are increased. Another important achievement is also illustrated: the class accuracy variance of the classification is now within the 34.8% to 90.8% interval; while in the previous work [20] it was between 22.6% and 87.4%. This may be attributed to the higher number of available training and testing samples as result of the BTI imputation, which increased the used statistical power of the dataset and improved the classification performance of the RF (which was the case for the other two classifiers as well).

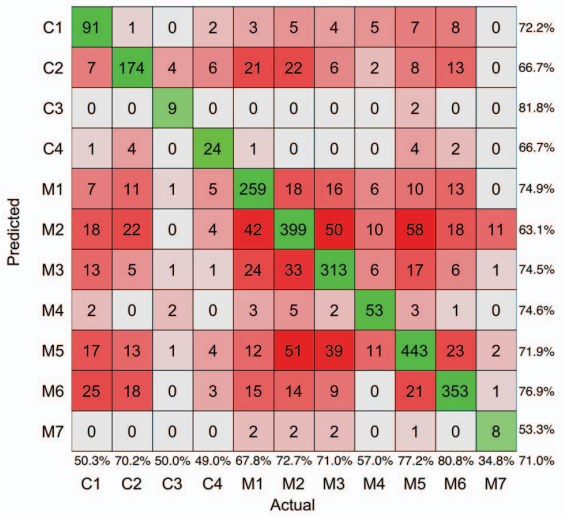

Fig. 5. Confusion matrix illustrating the RF classification results for the 11 classes, after employing BTI with continuous value coding. The batch includes 7 military ('M1' – Multi-function, 'M2' – Battlefield, 'M3' – Aircraft, 'M4' – Search, 'M5' – Air Defence, 'M6' – Weapon and 'M7' – Info) and 4 civil classes ('C1' – Maritime, 'C2' – Airborne Navigation, 'C3' – Meteorological, 'C4' – Air Traffic Control).

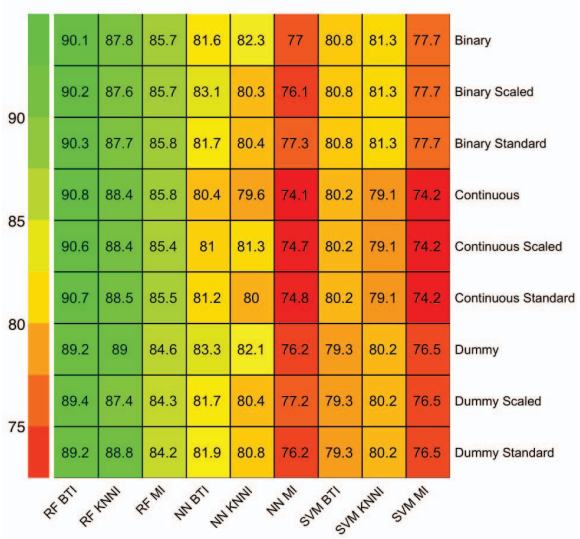

Fig. 6. Classification performance results of the classifiers (RF, NN and SVM) in the case of 2 classes after employing three different imputation techniques (BTI, KNNI and MI), for the three different groups of coding (binary, continuous and dummy). The colour scale to the left shows the achieved accuracy percentile.

The results shown in Fig. 6 and Fig. 7, illustrate that the categorical coding (binary and dummy) doesn't have drastic impact on the classification. However, the continuous coding appears to be more effective (1% in the 2 class case and 5% in the 11 class case). For the 2 class solution, the best result (90.80%) was obtained when combining the RF classifier with the BTI and continuous coding, which is also the case for the 11 class classification, when the same configuration achieved again the best accuracy (71.0%). The high number of patterns and the small number of features seems to provide a good generalization model that fits the test set.

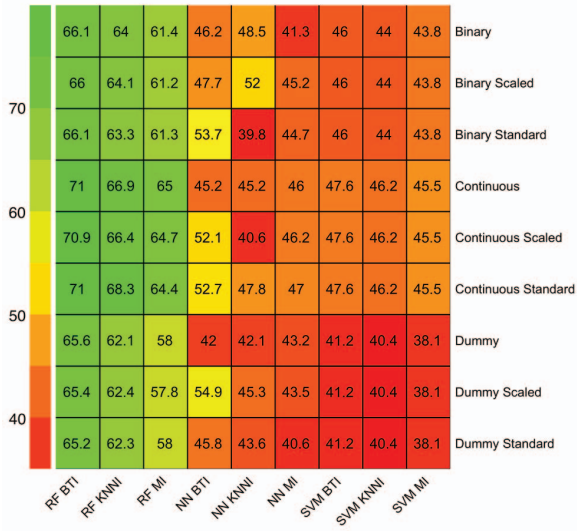

Fig. 7. Classification performance results of the classifiers (RF, NN and SVM) in the case of 11 classes after employing three different imputation techniques (BTI, KNNI and MI), for the three different groups of coding (binary, continuous and dummy). The colour scale to the left shows the achieved accuracy percentile.

In Fig. 6 and Fig. 7, the SVM columns have the same results for the scaled and standardized data, since the algorithm performs internally the two operations before the classification.

In 98 out of 108 comparisons (54 comparisons for each experiment (2-class and 11-class classification): same classifier and same coding against different imputation method), BTI has obtained the best accuracy (the numbers of the first column of each classifier in Fig.6 and Fig.7). For the rest of the cases: 8/108 were for KNN and 2/108 for MI.

VII. Conclusion

Three imputation models were employed for dealing with a large number of missing data in the available radar signal dataset. The use of MI on samples with up to 60% of missingness enabled us to double the data subset used in our previous work [20] and the use of BTI and KNNI tripled the number of patterns available for the classification. Subsequently, this larger dataset was used for supervised learning, when solving the radar signal classification and identification problem.

Subsequently, three classification approaches (NN, SVM, and RF) were implemented for recognition of generic radar data signal train pulse sources, and the classification was tested and validated in two experiments. The first set of simulations was for two classes and the second one for eleven output classes. The results were compared and critically analysed, showing overall improved accuracy when the classifier is trained on the larger subset with the bagged tree imputed values.

Future work that can further improve the classification accuracy will include investigation of the classification not only between the two 'super' classes '*military*' and '*civil*', but also investigation of the inner class separability and accuracy. For this purpose, appropriate statistical metrics will be introduced for assessing the methods' performance. The possibility of applying imputation methods to samples with higher than 60% missingness will be also investigated.

References

[1] C. Enders, Applied missing data analysis. New York: Guilford Press, 2010.

[2] J. Osborne, Best Practices in Data Cleaning, London: SAGE, 2013.

[3] G. E. Batista and M. C. Monard, "A Study of K-Nearest Neighbour as an Imputation Method," HIS Journal, vol. 87, pp. 251-260, 2002.

[4] G. Rahman, and I. Zahidul, "A decision tree-based missing value imputation technique for data pre-processing," Proceedings of the Ninth Australasian Data Mining Conference, vol. 121, pp. 41-50, 2011.

[5] M. Saar-Tsechansky, and P. Foster, "Handling missing values when applying classification models," J. of Machine Learning Research, vol. 8, pp. 1625-1657, 2007.

[6] M. A. Richards, Fundamentals of radar signal processing (2 ed.), USA: McGraw-Hill Education, 2014.

[7] N. Ibrahim, R. Abdullah, and M. Saripan, "Artificial neural network approach in radar target classification," J. of Computer Science, vol. 5 (1), pp. 23-30, 2009.

[8] Z. Zhang, X. Guan, and Y. He, "Study on radar emitter recognition signal based on rough sets and RBF neural network," IEEE Int. Conf. on Machine Learning and Cybernetics, vol. 2, pp. 1225-1230, 2009.

[9] C. Shieh, and C. Lin, "A vector neural network for emitter identification," IEEE Trans. on Antennas and Propagation, vol. 50 (8), pp. 1120-1127, 2002.

[10] Z. Yin, W. Yang, Z. Yang, L. Zuo, and H. Gao, "A study on radar emitter recognition based on SPDS neural network," Information Technology Journal, vol. 10 (4), pp. 883-888, 2011.

[11] M. Ahmadlou, and H. Adeli, "Enhanced probabilistic neural network with local decision circles: a robust classifier," Integrated Computer-Aided Engineering, vol. 17 (3), pp. 197-210, 2010.

[12] M. Gong, J. Zhao, J. Liu, Q. Miao, and L. Jiao, “Change Detection in Synthetic Aperture Radar Images Based on Deep Neural Networks,” IEEE Trans. on Neural Networks and Learning Systems, vol. 27 (1), pp. 125-138, 2016.

[13] A. Eryildirim, and O. Ibrahim, "Pulse Doppler radar target recognition using a two-stage SVM procedure," Aerospace and Electronic Systems, vol. 47 (2), pp. 1450-1457, 2011.

[14] Z. Xin, W. Ying, and Y. Bin, "Signal classification method based on support vector machine and high-order cumulants," Wireless Sensor Network, vol. 2 (1), pp. 48-52, 2010.

[15] Y. Yang, J. Lei, W.Zhang, and C. Lu, "Target classification and pattern recognition using micro-Doppler radar signatures, " Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing on IEEE, pp. 213-217, 2006.

[16] A. Karatzoglou, D. Meyer, and K Hornik, "Support vector machines in R," J. of Statistical Software, vol 15 (9), pp. 1-28, 2006.

[17] L. Breiman, "Random forests," Machine learning, vol. 45 (1), pp. 5-32, 2011.

[18] O. Gislason, J. Benediktsson, and J. Sveinsson, "Random forests for land cover classification," Pattern Recognition Letters, vol. 27 (4), pp. 294-300, 2006.

[19] G. Ridgeway, "Generalized Boosted Models: A guide to the gbm package," Version 2.1 (2015). Available from: http://cran.r-project.org/web/packages/gbm/index.html.

[20] I. Jordanov and N. Petrov, “Sets with Incomplete and Missing Data – NN Radar Signal Classification,” IEEE WCCI’14 World Congress on Computational Intelligence, Beijing, China, pp. 218-225, 2014.

[21] B. Frénay and M. Verleysen, “Classification in the presence of label noise: a survey,” IEEE Trans. on Neural Networks and Learning Systems, vol. 25 (5), pp. 845-869, 2014.

[22] A. Feelders, "Handling missing data in trees: surrogate splits or statistical imputation?," Principles of Data Mining and Knowledge Discovery. Springer Berlin Heidelberg, pp. 329-334, 1999.

[23] M. Templ, A. Kowarik, and P. Filzmoser, “Iterative stepwise regression imputation using standard and robust methods,” J. of Computational Statistics and Data Analysis, vol. 55, pp. 2793-2806, 2011.

[24] M. Templ, A. Alfons, and P. Filzmoser, “Exploring incomplete data using visualization tools,” J. of Advances in Data Analysis and Classification, vol. 6 (1), pp.29-47, 2012.

[25] S. Verboven, K. Branden, and P. Goos, “Sequential imputation for missing values,” Computational Biology and Chemistry, vol. 31 (5), pp. 320-327, 2007.

[26] P. Ellis, The Essential Guide to Effect Sizes: An Introduction to Statistical Power, Meta-Analysis and the Interpretation of Research Results. United Kingdom: Cambridge University Press, 2010.