Classification of Ford Motor Data

Joerg D. Wichard

***Abstract*— In this work we apply mixed ensemble models in order to build a classifier for the Ford Classification Challenge. We build feature vectors from the data sequences in terms of first order statistics, spectral density and autocorrelation. Our model selection scheme is a mixture of cross-validation and bagging. The outcome is an ensemble model, that consits of several different models trained on random subsamples of the entire data set.**

I. Introduction

The *Ford Classification Challenge* [1] is part of the WCCI 2008 Competition Program [2] and was motivated by an automotive application. The given task is the classification of finite data sequences, which includes also data preprocessing and generation and selection of feature vectors. Our method consists of three steps:

- Data preprocessing
- Generating and selecting feature vectors
- Building ensemble based classification models

II. Data Preprocessing

The *Ford Classification Challenge* [1] consists of two data sets that include data samples from an automotive subsystem. The data was collected in batches of $N = 500$ samples per diagnostic session, splitted in a training, a validation and a test set. The size of training, validation and test set is listed in Table I. The training set provided also the classification labels, wherein +1 indicates that a specific symptom exists and -1 indicates that the symptom does not exist. Later during the competition, the validation labels were also published on the competition website [1].

Name	Training	Validation	Test
Ford A	3271	330	1320
Ford B	3306	330	810

TABLE I

The size of training, validation and test set for the Ford Classification Challenge.

In order to generate proper feature vectors for the classification task we had to scale the data. Let $\vec{x} = \{x_i\}_{i=1,\ldots,N}$ denote a sampled time series from a diagnostic session. We assume, that the $\vec{x}$ sequence represents a set of uniformly-spaced time-samples of some measured signal x(t), where t represents time.

We removed the mean from the each series and scaled it:

$$\vec{x}_s = \frac{\vec{x} - \hat{x}}{s_x}, \tag{1}$$

wherein s_x is given by the sum of the absolute values

$$s_x = \sum_{i=1}^{N} \|x_i\| \tag{2}$$

and $\hat{x}$ is the mean over time

$$\hat{x} = \sum_{i=1}^{N} x_i \ . \tag{3}$$

We further define the $\sigma(\vec{x})$ as the variance of $\vec{x}$.

III. Feature Selection

Let $spec(\vec{x}_s)$ denote the the spectral density and $acf(\vec{x}_s)$ denote the autocorrelation of the scaled time series (see [3] for definitions). We build preliminary feature vectors $\vec{y}$ from the time series in the following way

$$\vec{y} = (spec(\vec{x}_s)_{\{1,\ldots,40\}}, acf(\vec{x}_s)_{\{1,\ldots,60\}}, \hat{x}, s_x, \sigma(\vec{x})), \tag{4}$$

wherein we included the first 60 values of the autocorrelation and the first 40 values of the spectral density.

We used a feature selection approach to decide which parts of the spectral density and the autocorrelation should be taken into the final feature vectors.

Our feature selection approach follows in principle the method of variable importance as proposed by Breiman [4]. The underlying idea is to select descriptors on the basis of the decrease of classification accuracy after the permutation of these descriptors. Briefly, an ensemble of classification models is built, which uses all descriptors as input variables and the classification accuracy on an hold out data set is calculated. In a second step, the same is done after the successive permutation of each descriptor. The relative decrease of classification accuracy of the permutated descriptor compared to the unchanged case is the *variable importance* following the idea that the most discriminative descriptors are the most important ones ((see Breiman [5]). This is done several times and the mean variable importance together with variance is calculated.

The spectral density for higher frequencies is almost zero, so only the low frequency part is useful for classification and for calculating the variable importance so we decided to use only the first 40 values for the preliminary feature vectors. The autocorrelation is not vanishing for higher values of the time lag, so there is no natural cutoff. We had to make a choice and we decided to use the first 60 values into the preliminary feature vectors. In Figure 1 the variable importance is shown for the preliminary feature vectors on both data sets (*Ford A* and *Ford B*). It turned out, that in both cases the first values of the autocorrelation are the most discriminative descriptors. We decided to use only the most discriminative descriptors

Joerg D. Wichard is with the FMP Berlin, Molecular Modelling Group, Robert-Roessle-Str. 10, 13125 Berlin, Germany, (email: wichard@fmp-berlin.de)

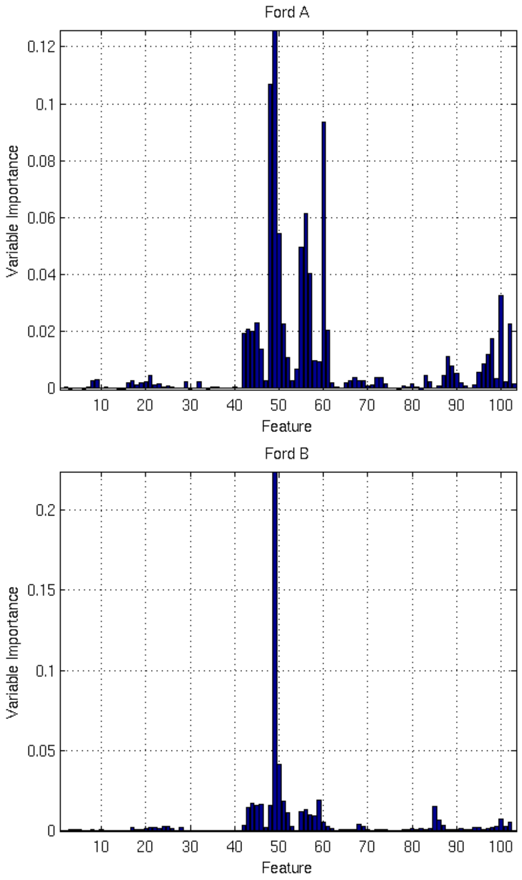

Fig. 1. The variable importance for the preliminary feature vectors on on both data sets (*Ford A* at the top, *Ford B* at the bottom). The first 40 values are the low frequency part of the spectral density, the next 60 values are the first values of the autocorrelation. The last three values are $\hat{x}, s_x, \sigma(\vec{x})$ as defined in Section II.

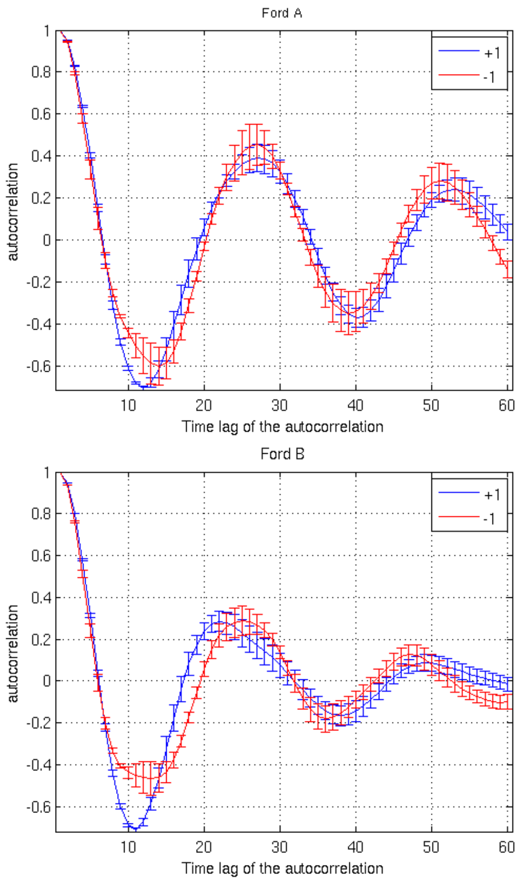

Fig. 2. The mean autocorrelation for both data sets (*Ford A* at the top, *Ford B* at the bottom). The positive labeled data is plotted in blue, the negative labeled data in red, together with the variance of the data shown as error bars.

that had a mean variable importance above the variance. Only 31 descriptors were left for *Ford A* and 30 *Ford B*. In Figure 2 we plotted the mean autocorrelation function of the two training data sets separately for the two classes (+1 and -1) with the variance as error bars. It is obvious, that the first two zero-crossings and the values around the first minimum are the most discriminative descriptors, as pointed out by the feature selection method descried above.

IV. Classifier Ensembles

The average output of several different models $f_i(\mathbf{x})$ is called an ensemble model

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{K} \omega_i f_i(\mathbf{x}), \tag{5}$$

wherein we define that the model weights ω_i sum to one $\sum_{i=1}^{K} \omega_i = 1$. There are several ways to define the model weights (see Perrone et al. [6] or Hashem et al. [7]), but we decided to use uniform weights with $\omega_i = 1/K$ for the sake of simplicity and not to run into over-fitting problems.

The central feature of the ensemble approach is the generalization ability of the resulting model. In the case of regression models (with continuous output values) it was shown, that the generalization error of the ensemble is in the average case lower than the mean of the generalization error of the single ensemble members (see Krogh 1995 [8]).

A. Model Selection

Our model selection scheme is a mixture of bagging [9] and cross-validation. *Bagging* or *Bootstrap aggregating* was proposed by Breiman [9] in order to improve the classification performance by combining classifiers trained on randomly generated subsets of the entire training sets. We extend this approach by applying a cross-validation scheme for model selection on each subset and after that we combine the selected models to an ensemble. In contrast to classical cross-validation, we use random subsets as cross-validation folds.

In K-fold cross-validation, the data set is partitioned into K subsets. Of these K subsets, a single subset is retained as the validation data for testing the model, and the remaining K - 1 subsets are used for model training. The cross-validation process is then repeated K times with each of the K subsets used only once as the validation data. The K results from the folds then can be averaged to produce a single estimation.

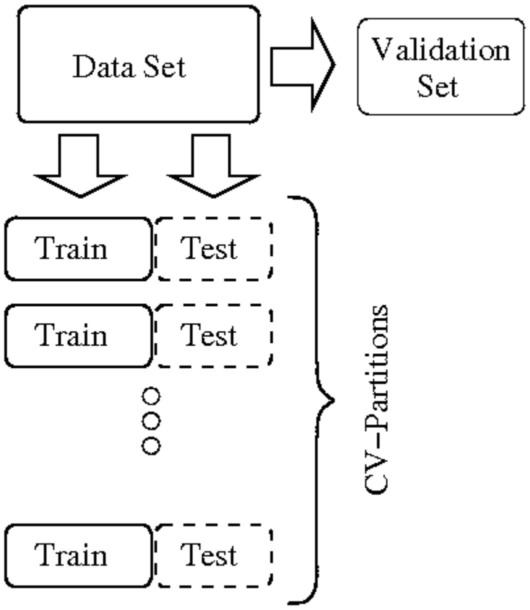

Fig. 3. For every partition of the cross-validation, the data is divided in a training and a test set. The performance of each ensemble model was assessed on validation set which was initially removed and never included in model training.

If we lack relevant problem-specific knowledge, cross-validation methods could be used to select a classification method empirically [10]. This is a common approach because it seems to be obvious that no classification method is uniformly superior, see for example Quinlan [11] for a detailed study. It is also a common approach to select the model parameters with cross-validation [12]. The idea to combine the models from the K cross-validation folds (stacking) was described by Wolpert [13].

We suggest to train several models on each CV-fold, to select the best performing model on the validation set and to combine the selected models from the K-folds. If we train models of one type but with different initial conditions (for example Neural Networks with different numbers of hidden neurons) then we find proper values for the free parameters of the model. We could extend that be combining models from different classes in order to increase the model diversity. We call this a *heterogeneous ensemble* or *mixed ensemble* and applied this method effectively to several problems [14]–[17].

Our model selection scheme works as follows: For the K-fold CV the data is divided K-times into a *training set* and a *test set*, both sets containing randomly drawn subsets of the data without replications. The size of each test set was 50% of the entire data set. In every CV-fold we train several different models with a variety of model parameters (see Section IV-C for an overview of the models). In each fold we select only one model to become a member of the final ensemble (namely the best model with respect to the test set). This means, that all models have to compete with each other in a fair tournament because they are trained and validated on the same data set. The models with the lowest classification error (the highest accuracy) in each CV-fold are taken out and added to the final ensemble, receiving the weight $\omega_i = \frac{1}{k}$. All other models in this CV-fold are deleted.

	predicted class +1	predicted class -1
real class +1	true positive (tp)	false negative (fn)
real class -1	false positive (fp)	true negative (tn)

TABLE II

THE CONFUSION MATRIX FOR A BINARY CLASSIFICATION PROBLEM.

B. Error Measures

We used the *accuracy* as defined in Eq. 6 in order to train and to compare the different classification models. Therefore we have to define the four possible outcomes of a classification that can be formulated in a 2×2 confusion matrix, as shown in Table II. The accuracy is defined as the ratio of the correct classified samples:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}. \qquad (6)$$

Another error measure is the false positive rate (FP-Rate), which is defined as the proportion of negative instances that were erroneously reported as being positive:

$$\text{FP-Rate} = \frac{fp}{tn + fp}.$$

C. Classification Models

In this section we give a short list of the model classes that we used for ensemble building. All models belong to the standard collection of machine learning algorithms for classification and regression tasks so details can be found in the textbooks like for instance Hastie et al. [18]. The implementation of these models in an open source MATLAB toolbox is available on-line [19] and allows the integration of user defined classification algorithms. A detailed description of our model classes was given recently [20]. The models that we use in our model selection scheme were:

- Linear Discriminant Analysis (LDA)
- Penalized Discriminant Analysis (PDA)
- Multi Layer Perceptron (MLP) with one or two hidden layers and randomly drawn number of neurons
- Support Vector Machines with RBF-kernels
- Classification and regression trees (CART)

V. Results

We applied our ensemble building approach to build a classification model for the two data sets from the *Ford Classification Challenge* [1]. The initial training set was used to train the model as described above. During the development phase (before the validation set labels were revealed) we tried different feature vectors and made our choice as described in Eq. 4. We further decided to use 101 cross-validation folds[1] in order to train the ensemble for the final submission, where the ratio of training/test samples

[1] A smaller number of cross-validation folds is possible, but a larger number cannot degrade the result.

Name	Accuracy	FP-Rate
Ford A (CV)	0.953 ± 0.002	0.041 ± 0.001
Ford B (CV)	0.946 ± 0.004	0.047 ± 0.001
Ford B (Orig.)	0.794	0.172

TABLE III

THE VALIDATION RESULTS FROM A 10-FOLD CROSS-VALIDATION RUN FOR THE TWO DATA SETS. WE FURTHER SHOWED THE RESULT OF THE ORIGINAL VALIDATION SET FOR THE FORD B DATA.

was 50/50 (see Section IV-A). After the disclosure of the validation labels, the validation set was combined with the training set in order to enlarge the set of labeled training samples for the final training.

From this enlarged training set, we generated small labeled validation sets of 20% size, that were use for a 10-fold cross-validation to estimate the expected classification error of our final model. It is worth to mention, that the cross-validation for the Ford B data set leads to relative good results while the accuracy for the original validation set is below average. The results of the cross-validation runs are reported in Table III.

Following the description on the challenge website [1] the data samples of hidden classification were collected under noisy conditions, while the data samples of known classification were collected in typical operating conditions. So training and operation conditions are different, which leads to a significant reduction of classification accuracy. In a final step, the validation set was combined with the training set and the model was trained for the submission. The ensembles for the final submission were build with 101 cross-validation folds, so each ensemble consits of 101 different models. For the *Ford A* data set the resulting ensemble consists of 41 SVMs with RBF-Kernels and 60 MLPs and for the *Ford B* data set we have 61 SVMs with RBF-Kernels and 40 MLPs.

VI. CONCLUSIONS

We successfully generated descriptors to describe the times series and applied a feature selection scheme in order to shrink the feature vectors. Our model selection approach selected and combined mostly MLPs and SVMs and the resulting ensemble models performed quite well on a 10-fold cross validation.

ACKNOWLEDGMENT

The authors would like to thank the members of the Molecular Modelling Group at FMP Berlin.

REFERENCES

[1] M. Abou-Nasr and L. Feldkamp, “Ford Classification Challenge,” http://home.comcast.net/˜nn-classification.

[2] “IEEE World Congress on Computational Intelligence (WCCI),” http://www.wcci2008.org/.

[3] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge UK: Cambridge University Press, 1997.

[4] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] ——, “Arcing classifiers,” *The Annals of Statistics*, vol. 26, no. 3, pp. 801–849, 1998.

[6] M. P. Perrone and L. N. Cooper, “When Networks Disagree: Ensemble Methods for Hybrid Neural Networks,” in *Neural Networks for Speech and Image Processing*, R. J. Mammone, Ed. Chapman-Hall, 1993, pp. 126–142.

[7] T. Hashem and B. Schmeiser, “Improving model accuracy using optimal linear combinations of trained neural networks,” *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 792–794, 1995.

[8] A. Krogh and J. Vedelsby, “Neural network ensembles, cross validation, and active learning,” in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. The MIT Press, 1995, pp. 231–238.

[9] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[10] C. Schaffer, “Selecting a classification method by cross-validation,” in *Fourth Intl. Workshop on Artificial Intelligence & Statistics*, January 1993, pp. 15–25.

[11] J. Quinlan, *Comparing connectionist and symbolic learning methods*. MIT Press, 1994, vol. I, pp. 445–456.

[12] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[13] D. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, pp. 241–259, 1992.

[14] J. Wichard, M. Ogorzałek, and C. Merkwirth, “Detecting correlation in stock markets,” *Physica A*, vol. 344, pp. 308–311, 2004.

[15] A. Rothfuss, T. Steger-Hartmann, N. Heinrich, and J. Wichard, “Computational prediction of the chromosome-damaging potential of chemicals,” *Chemical Research in Toxicology*, vol. 19, no. 10, pp. 1313–1319, 2006.

[16] J. Wichard, “Model selection in an ensemble framework,” in *Proceedings of the IEEE World Congress on Computational Intelligence*, Vancouver, Canada, 2006, pp. 2187 – 2192.

[17] J. Wichard and M. Ogorzałek, “Time series prediction with ensemble models applied to the CATS benchmark,” *Neurocomputing*, vol. 70, no. 13-15, pp. 2371–2378, 2006.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. Springer-Verlag, 2001.

[19] J. Wichard and C. Merkwirth, “ENTOOL - A Matlab toolbox for ensemble modeling,” http://www.j-wichard.de/entool/, 2007. [Online]. Available: http://www.j-wichard.de/entool/

[20] J. Wichard, H. Cammann, C. Stephan, and T. Tolxdorff, “Classification models for early detection of prostate cancer,” *Biomedicine and Biotechnology, Special Issue of the FBIT 2007*, January 2008.