Improvement of Information Fusion Based Audio Steganalysis

Christian Kraetzer and Jana Dittmann
Otto-von-Guericke University of Magdeburg, PO Box 4120, 39016 Magdeburg, Germany
Contact email: christian.kraetzer@iti.cs.uni-magdeburg.de

ABSTRACT

In the paper we extend an existing information fusion based audio steganalysis approach by three different kinds of evaluations: The first evaluation addresses the so far neglected evaluations on sensor level fusion. Our results show that this fusion removes content dependability while being capable of achieving similar classification rates (especially for the considered global features) if compared to single classifiers on the three exemplarily tested audio data hiding algorithms. The second evaluation enhances the observations on fusion from considering only segmental features to combinations of segmental and global features, with the result of a reduction of the required computational complexity for testing by about two magnitudes while maintaining the same degree of accuracy.

The third evaluation tries to build a basis for estimating the plausibility of the introduced steganalysis approach by measuring the sensibility of the models used in supervised classification of steganographic material against typical signal modification operations like de-noising or 128kBit/s MP3 encoding. Our results show that for some of the tested classifiers the probability of false alarms rises dramatically after such modifications.

1. MOTIVATION AND INTRODUCTION

Steganalysis, as the technique to detect hidden communication channels in media files or streams, is one of a number of important techniques to establish trust in media data. Like other such techniques (e.g. source identification as in digital camera forensics[21]) it becomes increasingly a hot topic in environments where a high level of trust in media data is required, such as high security network areas or secure long term archiving. Especially in the latter case an archiving of malware or hidden channels might have yet unforeseeable consequences for the trust in complete digital archives in the future.

From previous work applying information fusion in steganalysis it is known that fusion is a good method to improve the performance of a steganalysis approach in practical evaluations in terms of universality (Kharrazi et al.[1]) or detection performance (Pevny and Fridrich[2], Kraetzer and Dittmann[3]), and thereby the overall value of steganalysis as a security mechanism. This paper extends previous considerations from Kraetzer and Dittmann[3] on the application of information fusion in audio steganalysis by following three closely related test goals:

A. Completion of the information fusion based audio steganalysis approach presented in Kraetzer and Dittmann[3] by practical observations on sensor level fusion
B. Integration of global features into the fusion based steganalysis approach, aiming for more accurate decisions derived in multi-level information fusion
C. Verification of the plausibility of the introduced steganalysis approach for two common audio processing operations (MP3 encoding and de-noising)

For addressing the first goal, additional/alternative source sensors are implemented in software to accompany the original audio signal. This idea is similar to work by Ru et al.[5], Ozer[4] et al. and Avcibas[6], where signal processing operations (de-noising[4,6] and linear predictive coding[5]) are used to generate an assumably unmarked version of the signal as a reference signal in non-blind steganalysis. Our approach uses the same basic assumption that by signal processing an alternative version of a stego-file can be created where the hidden message can no longer be retrieved. In contradiction to the previous approaches by Ru et al.[5], Ozer[4] et al. and Avcibas[6] the newly generated alternative signals are not used as reference. Instead we fuse original and alternative signal on sensor level, to generate our alternative source sensor output. The test results achieved here using three exemplarily selected data hiding algorithms show that the content influence is reduced dramatically, while the subsequent classifications show similar classification accuracies. Based on the results we believe that by employing content removing operations in fusion based steganalysis the gap between universality enhancing and application specific approaches might be reduced.

For the second test goal of the paper, the feature extractor used in Kraetzer and Dittmann[3] (AAST, the AMSL Audio Steganalysis Toolset) is re-evaluated and enhanced here for the possibilities of generating useful global features for steganalysis. The newly generated global features are then incorporated into the fusion based steganalysis approach and

their impact to complexity and classification performance is evaluated, showing that they result in similar classification accuracies while at the same time dramatically reducing the time required for the computation of the classification.
For the third goal, the verification of the plausibility of the introduced steganalysis approach, models generated for steganographic algorithms in the training phase of supervised classification are used in testing to classify the output of common (non-malicious) audio signal modifications (de-noising and MP3 compression). This is done to measure the error rates achieved, to show which impact those signal modifications have especially on the false positive rates. Thereby the plausibility of the models used in this steganalysis approach in terms of sensibility against other kinds of signal modifications is verified. The results show, that indeed the risk of false alarms is increased in some cases to 100% by performing non-malicious signal modifications.

The rest of this paper is structured as follows: section 2 summarizes the complete test scenario design for test goals A, B and C, including the test setup. Section 3 describes the test results for all three test goals identified above, while section 4 summarizes our work and highlights directions for future work.

2. THE TEST SCENARIO

Based on the goals defined for this work, this section summarizes the complete test scenario (consisting of the test design and a description of the test setup).

2.1. Test Design

The general approach within this paper to address all three test goals identified in section 1 is the systematic usage of statistical pattern recognition methods provided by the renown data mining suite Weka[22]. The necessary feature extraction for the involved classification tasks is performed by applying our own audio feature extractor AAFE (see section 2.2.5) in a version especially enhanced for this paper by new features (especially global features). It is tested on material marked by three exemplarily chosen data hiding algorithms[3].
As an initial step prior to the evaluations for all three test goals, marked versions of the two test sets introduced in section 2.2.3 are generated using the three exemplarily chosen data hiding algorithms from section 2.2.1. In a second initialization step the two signal modification de-noising (by re-quantization to 8 Bit resolution) and MP3 encoding (see section 2.2.2) are used to generate modified versions of the marked and unmarked material for the evaluations of test goals A (sensor level fusion) and C (plausibility of the introduced steganalysis).
As a first evaluation step, the results of the analyses on the marked and unmarked material without fusion or signal modifications are presented for global and segmental features to act as a base for comparisons (for all three test goals) and to establish the required classifier quality estimations. The quality estimations are made based on the classifier quality Q defined in section 2.2.6. The results of these evaluations (using 10-fold stratified cross validation on the test set *aats389* as well as independent training with *aats389* and testing with *testset24* for the global as well as segmental features) are presented in section 3.1
In a second evaluation step, the impact of sensor level fusion (test goal A) using the fusion operator for this level described in section 2.2.7 is evaluated in section 3.2 for 10-fold stratified cross validation on *aats389* as well as independent training with *aats389* and testing with *testset24* for the global as well as segmental features.
Third, in section 3.3 the benefit of mixed level (or multi level) fusions is briefly evaluated (test goal B). While the results for the global features presented in this paper are in 3.1 and 3.2 derived by decision level observations (one decision is generated per file), the segmental features are evaluated in these chapters on a per frame basis (i.e. in match level observations). If the output of individual experts on the same type of features is used as input for an information fusion, then the fusion can be performed on the same level on which the experts work (e.g. match level for the segmental tests done here). If experts working on different kinds of features are joined, then the fusion has to be performed on the highest of the used levels. In our tests in section 3.3 this would therefore be the decision level. Here two exemplarily intra-level fusions (five best classifiers on global features and five best classifiers on segmentals for each algorithm) and one mixed- (or multi-)level fusion (the two best globals and the best segmental) are performed and compared with each other and the performance of the fest single classifiers involved in these tests.
In a fourth set of evaluations, the five best classifiers for each algorithm (selection of the algorithms is based on the quality function defined in section 2.2.6) are trained on the output of the de-noising and MP3 conversion run on a completely unmarked *aats389* and then used to verify (test) a completely unmarked *testset24* after the corresponding signal modification (test goal C).

2.2. Test Setup

The practical test setup includes three data hiding algorithms used for testing in test goals A,B and C, the additional signal modification operations required for sensor level fusion and the plausibility testing (test goals A and C), the audio test sets, pre-preprocessing, the feature extractor, the selected classifiers and quality function as well as the used fusion operators.

2.2.1. Data hiding algorithms used

In this paper a set of three data hiding algorithms (the steganography algorithm AMSL LSB stego, short *AS1* and the watermarking algorithms AMSL Spread Spectrum Watermarking *AW1* and Wasp *AW3* – for algorithm descriptions see Kraetzer et al.[19]) from some of our previous publications[3,19] is used here again for the generation of training and test data. Both classes of data hiding algorithms (steganography and digital watermarking) are present in the tests performed and can be compared in their statistical detectability. Each of the three algorithms is working in a different domain: *AS1* is a time domain LSB algorithm, *AW1* a frequency domain spread spectrum technique and *AW3* a wavelet domain algorithm.

The embedding is performed using exactly one predefined key ("UniversityOfMagdeburg") for the generation of the marked versions of the training and test files – i.e. in all files the message is embedded using the same key. All files used in the evaluations are marked by the data hiding algorithms with 100% capacity (message to be embedded is Goethes' "Faust" in an ASCII representation). No evaluations on test sets with reduced message lengths or an estimation of message lengths, such as presented e.g. by Fridrich et al.[20] or the impact of different key selection strategies[3], are considered here - these topics are reserved for further work.

Table 1: Algorithm parameterizations used

Alg.	Embedding domain	capacity used	message	fixed key
AS1 (AMSL LSB stego)	Time	100%	Goethes' "Faust"	UniversityOfMagdeburg
AW1 (AMSL Spread Spectrum Watermarking)	Frequency			
AW3 (Wasp)	Wavelet			

2.2.2. Additional signal modifications used

For the evaluation of test goals A (sensor level fusion) and C (plausibility of the introduced steganalysis) signal processing operations additional to the data hiding methods used (see section 2.2.1) have to be defined. These operations are a de-noising by re-quantization to 8 Bit resolution (and back to 16 Bit) and a MP3 encoding with 128kBit/s. The purpose of the re-quantization is two-fold: next to the MP3 conversion it is one of the two common signal modifications against which the classifier models are tested for the plausibility evaluation in test goal C. Additionally the output of this de-noising is used to generate by software the second sensor signal required for sensor level fusion (see section 2.2.7).

2.2.3. Audio test sets used in training and testing

For the tests performed here the two test sets (*aats389, testset24*) from Kraetzer et al.[3] are used again. The first set contains 389 PCM coded audio files with an average duration of 28.55s is used for evaluations based on 10-fold stratified cross validation as well as for training purposes in independent training and testing. The set *testset24* is used for the actual tests in independent training and testing, containing exactly one file per genre present in *aats389*. This second test set has a duration per file of approximately 30 seconds. Table 2 below summarizes the test sets used.

Table 2: Audio test sets used

Name of the test set	Syntactical properties	Number of files	Avg. duration
aats389	PCM16, 44.1kHz, stereo	389	28.55s
testset24	PCM16, 44.1kHz, stereo	24	30s

2.2.4. Pre-processing

Generally a windowing with rectangular, non-overlapping windows of window size 1024 samples is performed prior to feature extraction. For each file 20 windows per channel are used for the evaluations, since all used test files are stereo this results in 40 frames (equal to about 0.9s of audio material per file).

2.2.5. The feature extractor used

For the evaluations in this paper an enhanced version of the AMSL Audio Steganalysis Toolset (AAST[3]) is used. The main modification was the re-implementation and enhancement of the integrated feature extractor AMSL Audio Feature Extractor (AAFE). The new version (v.2.0.5) inherits the time-, frequency- and Mel-cepstrum domain based intra-window features from its predecessor (v.1.0.4) and extends them by 3 new ones in time domain (the zero crossing rate[9], the energy[10] and RMS-amplitude[11]), 520 new features in frequency domain (spectral centroid[12], spectral flux[13], spectral roll-off[14], spectral bandwidth[10], spectral smoothness[14], spectral irregularity[11], spectral entropy[11] and base frequency[15], as well as a 512 frequency component histogram[16]) and 26 in Mel-cepstrum domain (MFCCs and FMFCCs[18] on the second order derivative of the audio signal[17]). The most significant addition to features of the previous version of AAFE are the MFCCs and FMFCCs on the second order derivative of the audio signal, which have been added due to their good results achieved by Liu et al.[17]. The extractor therefore computes now **590 intra-frame features**: 8 in time domain ($sf_{zero\_cross\_rate}$, sf_{energy}, $sf_{entropy}$, $sf_{lsbratio}$, $sf_{lsbflipprate}$, $sf_{RMS\_amplitude}$, sf_{mean}, sf_{median}), 530 in frequency domain ($sf_{sp\_centroid}$, sf_{sp_flux}, $sf_{sp\_rolloff}$, sf_{sp_bw}, $sf_{sp\_smoothness}$, $sf_{sp\_irregularity}$, $sf_{sp\_entropy}$, $sf_{sp\_base\_freq}$, $sf_{Formant\_A1}$, …, $sf_{Formant\_singer}$, sf_{spec1},…,$sf_{spec512}$) and 52 in Mel-cepstrum domain (sf_{MFCC1}, ..., sf_{MFCC13}, sf_{FMFCC1}, ..., $sf_{FMFCC13}$, $sf_{d2MFCC1}$, ..., $sf_{d2MFCC13}$, $sf_{d2FMFCC1}$, ..., $sf_{d2FMFCC13}$).

Additionally **17 new inter-window features** are computed by AAFE v.2.0.5, as a means to reduce the computational complexity for the following classification operations. The following features are implemented: the total zero crossing rate and short time energy[10] as well as the averages of all nine time domain features and the arithmetic averages of the spectral entropy, spectral rolloff, spectral centroid, spectral bandwidth (threshold based), spectral irregularity and a spectral bandwidth which is center frequency based[13] (gf_{zcr_total}, gf_{zcr_AVE}, gf_{energy_AVE}, gf_{low_energy}, $gf_{entropy\_AVE}$, gf_{pitch_AVE}, gf_{median_AVE}, gf_{mean_AVE}, $gf_{lsb\_ratio\_AVE}$, $gf_{lsb\_flipping\_rate\_AVE}$, $gf_{rms\_amplitude\_AVE}$, $gf_{spectral\_entropy\_AVE}$, $gf_{spectral\_rolloff\_AVE}$, $gf_{spectral\_centriod\_AVE}$, $gf_{spectral\_bandwidth\_AVE}$, $gf_{spectral\_irregularity\_AVE}$ and $gf_{spectral\_bandwidth\_2\_AVE}$).

2.2.6. The selected classifiers and quality function

For the test goals A and B all 74 classifiers implemented in the current version of Weka (version 3.6.1[22]) with their default parameterizations are used. For a detailed description of the classifiers and their default parameters see the Weka manual or Hall et al.[22]. From this complete list of classifiers a subset of algorithms is selected for further evaluations on test goals B and C by usage of a quality function. The quality Q of a classifier is determined in this paper as a function of its classification accuracy (*accuracy*; defined here as the sum of all true positives and true negatives divided by the overall number of test samples, therefore a bigger value indicates better classification behavior while an accuracy of 50% in our two-class problem would be equal to the classifier randomly "guessing" at the class) and time required for the testing (*time*), as:

$$Q = \begin{cases} accuracy/time & if \quad accuracy > 50\% \\ 0 & if \quad accuracy = 50\% \end{cases}$$

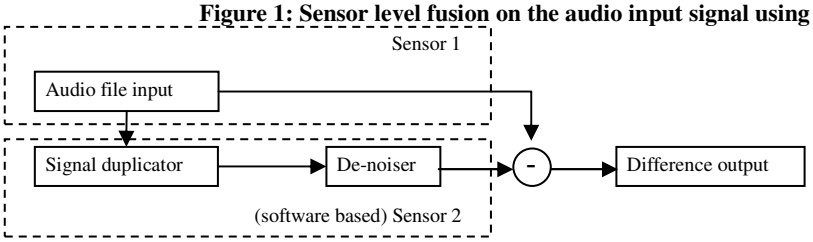
In case of accuracy=50% (i.e. the classifier is guessing at the result) Q is set to zero to avoid arithmetic problems. The *time* is determined by using the Wekas own time measurements.

The modeling of this quality function reflects the observation that in practical application a faster decision is sometimes much more valuable than a much slower but slightly more accurate answer.

2.2.7. Fusion operators

Two different types of fusion operators are used in this paper: one on sensor level and one in late fusion on match and decision level. For the first fusion operator figure 1 shows the process which takes the input of two sensors (the audio input device and a de-noised version of the signal – a kind of software based sensing; see section 2.2.2), combines (fuses) them by sample-wise subtraction and then outputs the difference signal, which is equivalent to the signal part removed by the de-noising. The sensor characteristics of the first sensor in figure 1 is therefore a simple file.read() function on a WAV audio file while the second sensor performs a file.read() on the same file followed by down quantization to 8 bit and requantization to 16 bit.

Figure 1: Sensor level fusion on the audio input signal using a de-noiser and subtraction



If the output of individual experts on the same type of features is used as input for an information fusion, then the fusion can be performed on the same level on which the experts work (e.g. match level for the segmental tests done here). If experts working on different kinds of features are joined, then the fusion has to be performed on the highest of the used levels – in this paper we perform in late fusion two exemplarily intra-level fusions and one mixed- (or multi-)level fusion by using as the second fusion operator in this paper a simple majority vote (on classifiers selected using the quality function described in section 2.2.6).

3. TEST RESULTS

This section introduces the test results achieved based on the test scenario described in section 2. First, as a base for comparisons for all three goals and to establish the required classifier quality estimations, the results of the analyses without fusion are presented for global and segmental computed features. In section 3.2 a sensor level fusion is evaluated for test goal A to show which impact content removing operations have on the classification process in steganalysis. In 3.3 for test goal B two selected intra-level and one mixed- (or multi-)level fusion are performed to evaluate the impact of late fusions (match- and decision level) on the classification accuracy. In the last part of this chapter, for test goal C an investigation into the plausibility of models trained for steganalysis against common signal modifications like MP3 encoding or de-noising are evaluated exemplarily.

3.1. Experimental comparison of global and segmental feature based results

Especially for test goal B, but also as a required point of reference for the other two test goals, we evaluate in this section the performance of the 74 classifiers in Weka using the global as well as the segmental features. Furthermore the best classifiers for each data hiding algorithm and feature type (global or segmental) are determined using the quality function *Q* introduced in section 2.2.6. Table 3 summarizes the results for the global features for 10-fold stratified cross validation on the set *aats389* as well as independent training with *aats389* and testing with *testset24*. The best results in each column are highlighted in bold script.

Table 3: Classification accuracies overview over all 74 WEKA classifiers (using: global features on 40 frames á 1024 samples per file in 10-fold stratified cross validation on *aats389* and training with *aats389* and testing with *testset24*)

	10-fold strat. cross valid.			**Training and testing**		
	AS1	***AW1***	***AW3***	***AS1***	***AW1***	***AW3***
Maximum achieved accuracy	**86.15%**	**92.68%**	**68.94%**	**57.29%**	**75.00%**	**59.38%**
Time duration (s)	598.7	603.8	617.3	222	191	180
Errors	13	14	14	13	13	13
50<=x<52%	24	8	11	55	9	16
52<=x<60%	32	1	9	**6**	2	**45**
60<=x<70%	2	1	**40**	0	34	0
70<=x<80%	0	6	0	0	**16**	0
80<=x<90%	**3**	24	0	0	0	0
x>=90%	0	**20**	0	0	0	0

The results presented in table 3 show the following: the highest achieved maximum classification accuracy in cross validation and training and testing is seen for *AW1* (92.68% and 75% respectively). *AW3* shows in average the second best result, while *AS1* ranks last – confirming our previous findings under similar test setups[3]. With an average duration a little bit over 600s for all tests the cross validation performs only 3-times slower than the training and test. In all tests

the same set of about 13 classifiers (in two cases 14) are not able to successfully generate a decision. The reasons for this are two fold: some classifiers terminate with an error stating that they have not enough memory to complete the task, some classifiers are terminated after 12 hours to keep the overall test duration limited (the time for thoses terminated is not included in the timings presented in the tables 3 and 5), others are cost sensitive classifiers which would not run without a cost file, which was not available for this test.
The lower six rows in table 3 are basically a histogram of how many classifiers achieved accuracies in the corresponding ranges. Again *AW1* shows the best performance: in cross validation for 20 classifiers an accuracy of more than 90% was achieved, with a maximum at 92.68% (*weka.classifiers.trees.LMT*). Generally the results in the training and testing setup are lower than in the cross validation, this seems to be due to the lower correlation between the test and training data in this case. The good results found for *AS1* in cross validation (*weka.classifiers.lazy.IB1* 84.87%, *weka.classifiers.lazy.IBk* 84.87% and *weka.classifiers.lazy.KStar* 86.15%) would not be confirmed in independent training and testing.

Table 4 identifies the best five classifiers for each algorithm and the global features. In contrast to our previous publications the best algorithms are here not determined by looking only on the classification accuracies achieved but instead by applying the quality function *Q* defined in section 2.2.6, basically determining the ratio between a classifiers accuracy and the time it requires for the decision. This is based on the observation that in practical application a faster decision is sometimes much more valuable than a much slower but slightly more accurate answer.
These classifiers identified in table 4 are used in the consecutive tests in sections 3.3 and 3.4.

Table 4: Identification of the five classifiers for each algorithm with the best quality value *Q*, determined in 10-fold stratified cross validation on *aats389* using global features (the value in brackets gives the value for *Q*)

	AS1	***AW1***	***AW3***
Max. quality *Q*	0.24	0.43377692	0.16866066
Best classifier	lazy.IBk (0.24)	trees.RandomTree (0.43)	rules.OneR (0.17)
2nd	lazy.IB1 (0.09)	trees.REPTree (0.42)	meta.FilteredClassifier (0.14)
3rd	meta.FilteredClassifier (0.06)	rules.OneR (0.41)	trees.DecisionStump (0.12)
4th	trees.DecisionStump (0.06)	trees.DecisionStump (0.38)	bayes.BayesNet (0.12)
5th	bayes.BayesNet (0.06)	misc.HyperPipes (0.36)	meta.AttributeSelectedClassifier (0.11)

The best classifier for *AS1* (*weka.classifiers.lazy.IBk* accuracy 84.87% at about 3.5s runtime) achieves about half the rating as the best classifier for *AW1* (*weka.classifiers.trees.RandomTree* 89.47% accuracy at about 2s runtime), simply because it takes about two times the time to reach a decision with a similar accuracy. The same is true for the best classifier for *AW3* (*weka.classifiers.rule.OneR* 60.29% accuracy at about 3.6s).

Similar to table 3, table 5 shows the summary of the classifier results for the global features for 10-fold stratified cross validation on *aats389* as well as training with *aats389* and testing with *testset24*.

Table 5: Classification accuracies overview over all 74 WEKA classifiers (using: segmental features on 40 frames á 1024 samples per file in 10-fold stratified cross validation on *aats389* and training with *aats389* and testing with *testset24*)

	10-fold strat. cross valid.			**Training and testing**		
	AS1	***AW1***	***AW3***	***AS1***	***AW1***	***AW3***
Maximum achieved accuracy	**94.19%**	**96.47%**	**70.36%**	**50.12%**	**89.48%**	**61.22%**
Time duration (s)	104467.5	86056.2	113349.6	30870	35852.05	54521.25
Errors	31	29	35	29	27	31
50<=x<52%	35	0	1	**45**	0	20
52<=x<60%	3	8	26	0	11	17
60<=x<70%	0	3	11	0	1	**6**
70<=x<80%	1	1	**1**	0	4	0
80<=x<90%	0	3	0	0	**31**	0
x>=90%	**4**	**30**	0	0	0	0

Again *AW1* shows the best detectability with 30 classifiers achieving an accuracy larger than 90% in cross validation here the segmentals seem to outperform the global features in terms of accuracy. Like already shown for the global features, the good results for four classifiers on *AS1* in cross validation could not be verified with independent training and testing.
If the results for the global and segmental features are compared directly, besides the slight overall increase in the classification accuracy two additional facts are noticeable: the number of classification algorithms which can not fulfill the classification task (row "error" in the tables) more than doubles and the time duration for the tests increase by at least two magnitudes. The former is due to the increased memory requirement for the segmental features (by factor 20 in comparison to the globals) and a corresponding increase of the number of "out of memory errors" from Weka, while the latter is due to the much more complex training and testing tasks at hand and thereby more timeouts at our 12h limit.

Table 6: Identification of the five classifiers for each algorithm with the best quality value *Q*, determined in 10-fold stratified cross validation on *aats389* using segmental features (the value in brackets gives the value for *Q*)

	AS1	***AW1***	***AW3***
Max. Quality *Q*	0.00221	0.01	0.00092
Best	rules.OneR (0.00221)	misc.HyperPipes (0.0100)	misc.HyperPipes (0.00092)
2nd	trees.RandomTree (0.00093)	trees.RandomTree (0.0097)	trees.RandomTree (0.00088)
3rd	trees.RandomForest (0.00043)	misc.VFI (0.0044)	meta.Vote (0.00065)
4th	lazy.Ibk (0.00041)	rules.OneR (0.0030)	meta.CVParameterSelection (0.00059)
5th	misc.HyperPipes (0.00022)	trees.RandomForest (0.0028)	rules.ZeroR (0.00058)

Table 6 identifies similar to table 4 the best five classifiers for each algorithm and the segmental features for use in sections 3.3 and 3.4. Since the classification times for the segmental features are – due to their much lager and more numerous feature vectors – longer than for the globals, the value for *Q* is much lower, even while similar accuracies are achieved (e.g. *AW1 weka.classifiers.misc.HyperPipes* at 83.89% and 85s run time). Summarizing those observations on the time behavior it has to be said that the usage of global features (based on the same input signal amounts) results in similar classification accuracies, while at the same time the probability for an algorithm to succeed the task (without an "out of memory error") increases and the overall computation time for the tests decreases (on the machine used for the test in this paper from 29h for one algorithm and all successful classifiers in cross validation down to 10 minutes).

3.2. Sensor level fusion results (test goal A)

After the previous section introduced as a required reference the results without fusion, here the results with the sensor level fusion introduced in section 2.2.7 are given. Table 7 summarizes the results for the global features for 10-fold stratified cross validation on *aats389* as well as training with *aats389* and testing with *testset24*.

Table 7: Sensor level fusion - classification accuracies overview (using: all 74 classifiers, global features on 40 frames á 1024 samples per file in 10-fold stratified cross validation on *aats389* as well as training with *aats389* and testing with *testset24*)

	10-fold strat. cross valid.			**Training and testing**		
	AS1	***AW1***	***AW3***	***AS1***	***AW1***	***AW3***
Maximum achieved accuracy	**92.88%**	**90.57%**	**66.55%**	**55.68%**	**69.57%**	**58.89%**
Time duration (s)	674.0	647.0	518.0	194.0	193.0	159.0
Errors	13	13	13	14	14	14
50<=x<52%	41	8	8	52	0	22
52<=x<60%	12	1	9	**8**	15	**38**
60<=x<70%	0	2	**44**	0	**45**	0
70<=x<80%	0	3	0	0	0	0
80<=x<90%	3	23	0	0	0	0
x>=90%	**5**	**24**	0	0	0	0

When comparing these results achieved using sensor level fusion with their counterparts without fusion in table 3 it can be stated that the results achieved by this rather crude sensor level fusion already give results which in terms of accuracy

close to the results presented in the previous section. For some algorithms (e.g. *AS1* and *AW1* in cross validation) the number of classifiers with better accuracies actually even outperforms those achieved without sensor level fusion.

Table 8: Sensor level fusion - classification accuracies overview (using: 74 classifiers, segmental features on 40 frames á 1024 samples per file in 10-fold stratified cross validation on *aats389* as well as training with *aats389* and testing with *testset24*)

	10-fold strat. cross valid.			**Training and testing**		
	AS1	***AW1***	***AW3***	***AS1***	***AW1***	***AW3***
Maximum achieved accuracy	**94.64%**	**75.68%**	**66.57%**	**50.53%**	**60.85%**	**54.20%**
Time duration (s)	75831.0	147868.7	118597.5	25186.0	73496.0	42201.6
Errors	32	33	33	30	28	31
50<=x<52%	31	0	0	**44**	1	23
52<=x<60%	4	8	**40**	0	36	**20**
60<=x<70%	2	5	1	0	**9**	0
70<=x<80%	1	**28**	0	0	0	0
80<=x<90%	0	0	0	0	0	0
x>=90%	**4**	0	0	0	0	0

While the results presented in table 8 still show significant classification results, it has to be admitted that their results are worse than in the case of the global features and sensor level fusion or their direct counterparts without fusion (see table 5). Nevertheless the basic idea of using sensor level fusion for content influence elimination in steganalysis seems to be a very promising one, which should be subject of further research.

3.3. Mixed level fusion (test goal B)

In this section the benefit of mixed level (or multi level) fusions is briefly evaluated, after section 3.1 already compares global and local features without fusion. Table 9 shows the results for two intra-level and one mixed- (or multi-)level fusion. In the first intra level fusion the decisions of the best five classifiers (based on *Q* - see table 4) per algorithm are joined on decision level and compared to the result for the best single classifier in this set of five. In the second fusion the same is done on match level for the segmental features.
The third fusion presented is a mixed level fusion, where the results of the best two classifiers on global features are joined with the best classifier on segmental features (after the results of the latter have been aggregated for each file by majority decision). The classifiers used for each algorithm in these evaluations are identified in tables 4 and 6. The best singles shown in the last row of table 9 are: for *AS1*: *weka.classifiers.rules.OneR* (segmental), *AW1*: *weka.classifiers.misc.HyperPipes*, *AW3* (segmental): *weka.classifiers.misc.HyperPipes* (segmental).

Table 9: Results for two intra-level and one mixed- (or multi-)level fusion (using: the best classifiers for each feature type on global and segmental features for 40 frames á 1024 samples per and training with *aats389* and testing with *testset24*)

	AS1	***AW1***	***AW3***
Fusion: five best (highest *Q*) globals (see table 4)	50.0%	**70.8%**	51.0%
Best accuracy for a single in these 5 classifiers	50.0%	69.8%	**56.3%**
Fusion: five best (highest *Q*) segmentals (see table 6)	50.0%	84.3%	53.0%
Best accuracy for a single in these 5 classifiers	**50.1%**	**86.8%**	**56.8%**
Fusion: two best (highest *Q*) globals and best segmental (see tables 4 and 6)	46.9%	**80.2%**	**57.3%**
Best accuracy for a single in these 3 classifiers	**50.1%**	71.6%	56.8%

The results show no positive effect for *AS1*, while for *AW1* the results for all fusions involving global features achieve a higher accuracy than the involved single classifiers. For *AW3* the intra-level show results lower than the best single expert in the group, but the mixed level fusion shows a better result.

3.4. Verification of plausibility (test goal C)

The five best classifiers for each algorithm (selection of the algorithms is based on the quality function defined in section 2.2.6) are trained on the output of the signal modifications run on a completely unmarked *aats389* and then used to verify (test) a completely unmarked *testset24* after the corresponding signal modification. Table 10 shows thereby the results of this test for the global features. A value of 100% in this table means that the complete test material in the evaluation was rightfully classified as unmarked by the corresponding data hiding algorithm. A value of 0% means that the classifier produced false alarms on every input sample. The best, 2nd best, etc. classifiers for each algorithm in these evaluations are resolved in table 4. Optimal (100%) values are barked in bold script.

Table 10: Classification results for the global features and the best five classifiers for each algorithm (files after signal modifications considered being unmarked; 40 frames á 1024 samples per and training with *aats389* and testing with *testset24*)

Signal modification	Classifier	*AS1*	*AW1*	*AW3*
MP3 encoding	best (see table 4)	56.82%	77.27%	63.64%
	2nd	56.82%	70.45%	95.45%
	3rd	**100%**	63.64%	0%
	4th	95.45%	77.27%	56.82%
	5th	**100%**	20.45%	72.73%
de-noising	best (see table 4)	29.55%	**100%**	79.55%
	2nd	29.55%	95.45%	**100%**
	3rd	**100%**	84.09%	**100%**
	4th	**100%**	90.91%	**100%**
	5th	**100%**	**100%**	**100%**

The results achieved in these evaluations are very interesting: While the de-noising operation output is in nine out of 15 cases tested here with the global features found 100% correct to be "not marked", in four other cases this value is still above 80%, while for two cases (the best and second best classifiers for *AS1*) the values are down to 29.55% equal to a rate of false alarms of about 70%. For the MP3 encoding the picture is even worse, with only two classifiers achieving 100% preciseness while all others show less than perfect results. One of the classifiers (the 3rd best for *AW3*) even shows a 100% false alarm rate.

Table 11 summarizes the results of the plausibility test for the segmental features. Again, a value of 100% would imply that no false alarms occurred, while 0% means that the classifier produced false alarms on every input sample. The best, 2nd best, etc. classifiers for each algorithm in these evaluations are resolved in table 6.

Table 11: Classification results for the segmental features and the best five classifiers for each algorithm (files after signal modifications considered being unmarked; 40 frames á 1024 samples per and training with *aats389* and testing with *testset24*)

Signal modification	*Classifier*	*AS1*	*AW1*	*AW3*
MP3 encoding	best (see table 6)	53.64%	51.14%	16.82%
	2nd	9.09%	76.93%	71.14%
	3rd	46.48%	56.48%	0%
	4th	6.93%	76.14%	0%
	5th	0%	75.57%	0%
de-noising	best (see table 6)	53.97%	99.88%	96.14%
	2nd	10.51%	83.76%	43.34%
	3rd	50%	65.07%	0%
	4th	1.75%	71.03%	0%
	5th	0%	77.57%	0%

In direct comparison to the results achieved for the global features is has to be stated that the segmental features perform significantly worse if it comes to plausibility against common signal modification operations. None of the 30 testes cases summarized in table 11 shows 100% preciseness, while eight cases show a false alarm rate of 100%.

4. SUMMARY AND CONCLUSIONS FOR FURTHER WORK

Within this paper the following four main points are observed in applied statistical pattern recognition on audio material, searching for traces left by the embedding of three selected data hiding algorithms:

- Sensor level fusion seems to be a good way of reducing the contend dependability in steganalysis. After the (rather crude) sensor level fusion performed in the tests (test goal A), the content influence is reduced dramatically, while the subsequent classifications show similar classification accuracies. Here further and more sophisticated methods for content removal/de-noising should be tested in future research.
- Fusion can be used either to generate a universal solution (like in Kharrazi et al.[1]) or a solution which is pretty customized for one application (goal) – here the increasing of the detection accuracy for a specific data hiding algorithm (test goal A). In this respect it would be interesting for future research to include sensor level fusion as considered here to try to close the gap between these two application goals for fusion. This would also make additional research in the area of definition of quality and cost functions for fusions a necessity.
- In the direct comparison between global and segmental computed features (test goal B) on the audio material for the purpose of steganalysis the complete list of Wekas 74 classifiers is tested here, leading to the observation that the usage of global features in the tests performed leads to similar classification accuracies at a much faster speed (10 minutes in comparison to 29h for one machine and one test through all 74 classifiers). Nevertheless segmental features would be required to detect the onset (time point) at which a data hiding embedding into an audio stream begin starts, something that should be considered for tests in future work.
- So far plausibility observations on classifier models trained in steganalysis have been much neglected. Nevertheless such test, as they are described and performed here (test goal C), are considered necessary by us to estimate the behavior of applied steganalysers in practical setups where signal modifying operations are common and might trigger unnecessary false alarms from the steganalyser.

The findings in this paper are for three reasons of relevance for environments which require a high trust assumption in media data, like e.g. secure long term archives: first it is shown that global features achieve similar classification accuracies as segmental features in the tests performed here, while reducing the required computation power tremendously – thereby the throughput of the security mechanism is increased. Second, the sensor level fusion tested here is basically a content removal or anonymisation of the data. This can be performed in an independent module inside the application (e.g. a secure archive) prior to the handing out of the data to a steganalysis module. Therefore the steganalyser does not obtain information about the content he has to check. Since the classification accuracies achieved in the tests here are as high as those without sensor level fusion this might be a welcome privacy enhancement to this security mechanism. Third, the tests showed a certain tendency for some steganalysis model to wrongfully flag non-maliciously modified material. In practical application, e.g. in a secure archive environment where some format conversions are foreseen, this indicates that the model and classifier combinations to be used have to undergo extensive field testing prior to large scale employment, to reduce the probability of false alarms.

Additional to the topics for further work already mentioned additional observations have to be spend on extending the test set (e.g. in terms of more algorithms) and evaluations with different embedding strengths and capacities[20], to allow for a better generalization of the findings. Also the impact of the key chosen for embedding should be re-evaluated[3]. Furthermore feature selection strategies[7] should be applied to the setup (and here especially to the segmental features) to allow on a per algorithm basis the identification of significant features and thereby a further optimization of the classification process. Also additional fusion operators (e.g. including accuracy based weighting strategies) besides the accuracy/time based selection and unweighted fusion should be evaluated in future research.
Furthermore the reasons for the good or bad performance of certain classifiers in the tests performed should be evaluate to gather on this way knowledge about the individual pattern recognition problems at hand.

ACKNOWLEDGEMENTS

The authors wish to thank Thomas Naumann for his work on AAFEv2 and the many discussions about its application to audio steganalysis. The work in this paper has been supported in part by the European Commission through the FP7 ICT Programme under Contract FP7-ICT-216736 SHAMAN. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REFERENCES

[1] M. Kharrazi, H. T. Sencar, and N. Memon: *Improving steganalysis by fusion techniques: A case study with image steganography*. In Transactions on Data Hiding and Multimedia Security *I*, Y. Q. Shi, ed., Springer LNCS 4300, 2006.

[2] T. Pevny and J. Fridrich: *Merging Markov and DCT Features for Multi-Class JPEG Steganalysis*. Proc. SPIE Electronic Imaging, Photonics West, January 2007.

[3] C. Kraetzer and J. Dittmann: *The Impact of Information Fusion in Steganalysis on the Example of Audio Steganalysis*. Proceedings of the Media Forensics and Security XI. Electronic Imaging Conference 7254, IS&T/SPIE 21th Annual Symposium, San Jose, CA, USA, January 18th-22nd, 2009.

[4] H. Ozer, I. Avcibas, B. Sankur, and N. Memon: *Steganalysis of audio based on audio quality metrics*. In SPIE Electronic Imaging Conf. On Security and Watermarking of Multimedia Contents, Jan. 20-24, Santa Clara, 2003.

[5] X.-M. Ru, H.-J. Zhang, and X. Huang: Steganalysis of audio: *Attacking the steghide*. In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18-21 August, 2005.

[6] I. Avcibas: *Audio steganalysis with content-independent distortion measures*. In IEEE Signal Processing Letters, Vol. 13, No. 2, February, 2006.

[7] C. Kraetzer and J. Dittmann: Impact of Feature Selection in Classification for Hidden Channel Detection on the Example of Audio Data Hiding. Proceedings of the 10th ACM Workshop on Multimedia and Security, September 22-23, 2008, Oxford, UK, 2008.

[8] A. Lang: *Audio Watermarking Benchmarking – A Profile Based Approach*. PhD Thesis. Otto-von-Guericke University Magdeburg, Department of Computer Science, 2008.

[9] Geoffroy Peeters: A Large Set of Audio Features for Sound Description. Ircam technical report, 2004.

[10] Sabine Keuser: Similarity Search on Musical Data. Master thesis at ETH Zurich, Institute of Information Systems, Database Research Group, 2002.

[11] Geoff Luck and Petri Toiviainen: Exploring Relationships between the Kinematics of a Singer's Body Movement and the Quality of Their Voice. Journal of interdisciplinary music studies, Spring/Fall 2008, volume 2, issue 1&2, pp. 173-186, 2008.

[12] Stefan Bindreiter: Project report Feature Extraction Audio B. University of Applied Sciences Hagenberg, Austria, 2005.

[13] Stefan Bindreiter: Audio gesteuerte Animationen. Master thesis, University of Applied Sciences Hagenberg, Austria, 2005.

[14] Thomas Riedel: Featureextraktion für die Klassifikation von Klangdaten, Diploma Thesis TU Chemnitz, Germany, 2003.

[15] Anto Zecevic: Ein sprachgestütztes Trainingssystem zur Evaluierung der Nasalität. PhD Thesis University of Mannheim, Germany, Dept. Mathematics and Computer Science, 2003.

[16] Robert Buchholz, Christian Kraetzer, Jana Dittmann: Microphone Classification Using Fourier Coefficients. Profeedings of 11th Information Hiding Darmstadt, Germany, June 7-10, 2009.

[17] Qingzhong Liu, Andrew H. Sung and Mengyu Qiao. Temporal Derivative Based Spectrum and Mel-Cepstrum Audio Steganalysis. IEEE Transactions on Information Forensics & Security, 2008.

[18] Christian Kraetzer and Jana Dittmann: Mel-Cepstrum Based Steganalysis for VoIP-Steganography; Proceedings of SPIE conference, at the Security, Steganography, and Watermarking of Multimedia Contents IV, IS&T/SPIE Symposium on Electronic Imaging, Jan. 28- Feb. 1st, 2007, San Jose, USA, 2007.

[19] C. Kraetzer and J. Dittmann: Pros and Cons of Mel-cepstrum based Audio Steganalysis using SVM Classification. Proceedings of Information Hiding 2007.

[20] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal: Quantitative Steganalysis of Digital Images: Estimating the Secret Message Length. ACM Multimedia Systems Journal, Special issue on Multimedia Security, Vol. 9(3), pp. 288-302), 2003.

[21] J. Lukas, J. Fridrich, and M. Goljan, Determining digital image origin using sensor imperfections, Proceedings of the SPIE International Conference on Image and Video Communications and Processing, vol. 5685, no. 1. SPIE, 2005.

[22] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.