Selection of Compressible Signals from Telemetry Data

Phillip Taylor*
The University of Warwick
Coventry, UK
phillip.taylor@warwick.ac.uk

Nathan Griffiths
The University of Warwick
Coventry, UK
nathan.griffiths@warwick.ac.uk

Alex Mouzakitis
Jaguar Land Rover Research
Coventry, UK

ABSTRACT

Sensors are deployed in all aspects of modern city infrastructure and generate vast amounts of data. Only subsets of this data, however, are relevant to individual organisations. For example, a local council may collect suspension movement from vehicles to detect pot-holes, but this data is not relevant when assessing traffic flow. Supervised feature selection aims to find the set of signals that best predict a target variable. Typical approaches use either measures of correlation or similarity, as in filter methods, or predictive power in a learned model, as in wrapper methods. In both approaches selected features often have high entropies and are not suitable for compression. This is of particular issue in the automotive domain where fast communication and archival of vehicle telemetry data is likely to be prevalent in the near future, especially with technologies such as V2V and V2X. In this paper, we adapt a popular feature selection filter method to consider the compressibility of signals being selected for use in a predictive model. In particular, we add a compression term to the Minimal Redundancy Maximal Relevance (MRMR) filter and introduce Minimal Redundancy Maximal Relevance And Compression (MRMRAC). Using MRMRAC, we then select features from the Controller Area Network (CAN) and predict each of current instantaneous fuel consumption, engine torque, vehicle speed, and gear position, using a Support Vector Machine (SVM). We show that while performance is slightly lower when compression is considered, the compressibility of the selected features is significantly improved.

KEYWORDS

Feature selection, Data compression, Data mining, Machine learning

ACM Reference Format:
Phillip Taylor, Nathan Griffiths, and Alex Mouzakitis. 2018. Selection of Compressible Signals from Telemetry Data. In *Proceedings of Mining Urban Data Workshop (MUD3).* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

We are living in ever smarter cities with increasing numbers of sensors producing larger amounts of data. Sensors are deployed in all aspects of city infrastructure, from train networks, to water and electrical grids. Collecting data from these sensors, storing and analysing it, presents a huge challenge for researchers. For example, a water provider may wish to detect leaks, and require analysis of data from thousands of sensors across the city. In the automotive domain, vehicles are set to communicate with traffic management systems (V2X) and other vehicles (V2V) to enhance both efficiency and safety [7, 17]. Existing communications infrastructure cannot satisfy the required bandwidth, necessitating new communication protocols and novel approaches to data analysis.

Supervised machine learning aims to build models that process inputs to output predictions for a target variable [18]. For example, a rail network may wish to learn a model that determines whether or not a train track is likely to fail in the near future. Such a model may take input data from sensors along the track along with telemetry from trains, and output an estimated likelihood of failure. Similarly, a highways agency may wish to locate pot-holes or predict traffic jams using road sensors and vehicle telemetry data. The number of possible input features in such domains is large, and many features may be irrelevant to the target variable or redundant with respect to other features. Both irrelevance and redundancy can cause increased model complexity or incorrect mappings being learned, which in turn lead to lower predictive performance. To overcome this, supervised feature selection can be used to find a subset of the features that are most related to the target variable (i.e. high relevance) but least related to each other (i.e. low redundancy) [4, 13]. Supervised feature selection has a bias for features that carry lots of information about a target variable, as these typically are the most relevant. This high information content often coincides with high entropies and poor compression, meaning these signals are likely to be expensive to communicate and difficult to store.

In this paper, we consider compression when performing feature selection, aiming to choose features with good predictive performance and high compression. Specifically, we adapt the Minimal Redundancy Maximal Relevance (MRMR) feature selection scheme [4, 13] to include a compressibility factor and introduce Minimal Redundancy Maximal Relevance And Compression (MRMRAC). We assess the performance of MRMRAC with respect to prediction accuracy and compression, demonstrating the algorithm on four predictive tasks using vehicle telemetry data.

2 RELATED WORK

Feature selection aims to find a subset of all possible features to reduce their number, while still sufficiently describing the data with respect to a particular task [2]. In unsupervised learning a task may be to group data samples in an efficient way, or in a supervised setting it may be to predict a given target variable. Feature selection for unsupervised learning typically aims to find the features that best capture differences between samples. This typically relies on assessing clustering properties, variance, or other discriminatory properties measured using heuristics.

*Contact author

MUD3, August 2018, London, UK

In supervised learning, which is the focus of this paper, the three main approaches are embedded, wrapper, and filter methods. Embedded methods perform feature selection as part of the learning algorithm [6]. For example, in decision tree induction, the choice of variable on which to split nodes can be seen as feature selection. The nodes used, and their associated features, are then the selected features. Also, trees in random forests may be removed if their estimated performance is poor, which may mean poor performing features are never used in the learned model.

In wrapper methods, feature subsets are assessed by estimating the performances of models that use them as inputs. This is done by comparing the performances of models learned using the same process and training samples, but with different sets of input features. Wrapper methods are typically computationally expensive, as they require a machine learning algorithm to build a model for each feature subset evaluation. Filter methods, on the other hand, generally assess the performances of feature subsets using heuristics that are typically less computationally expensive. Some commonly used heuristics include similarity measures such as Pearson's Correlation Coefficient (PCC) or Mutual Information (MI), which can be used to estimate both relevance and redundancy of a feature set [18].

In general, supervised feature selection can be represented as an optimisation problem [12],

$$\underset{S \subseteq X}{\operatorname{argmax}} \; \mathrm{P}(S, y), \tag{1}$$

where X is the set of all possible features, and $\mathrm{P}(S, y)$ estimates the performance of a subset of these features, S, with respect to predicting the target variable, y. Ideally, all possible subsets would be evaluated to find the feature set that provides the highest performance, but the number of possible feature subsets is $2^{|X|}$ and this exhaustive search is infeasible. In practice, therefore, a more efficient combinatorial search algorithm is applied.

Possibly the most common search strategy is the forward greedy search [12], which iteratively selects the feature, x, that satisfies,

$$\underset{x \in X \setminus S}{\operatorname{argmax}} \; P(S \cup \{x\}, y). \tag{2}$$

The search begins with no selected features, $S = \emptyset$, to which the feature with the highest individual performance is selected. This continues, selecting the feature that adds the highest performance. The search stops when a stopping criteria is met, such as when a given number of features are selected or if the performance score decreases after selecting a new feature.

Another approach is a backwards greedy search, in which features are iteratively removed from the set of selected features if this increases the estimated performance. It is typically more efficient to perform a forward search if the number of features is large, especially when using a wrapper approach. Other search algorithms used include randomised approaches such as genetic search or simulated annealing, using $P(S, y)$ as a fitness function [11].

Some filter methods avoid using traditional search algorithms. For example, feature redundancy can be discovered by clustering features by their similarities [10]. The feature from each cluster that is most relevant to the target can then be selected, or some composite feature generated. Similarly to this, principal components analysis extracts new features that capture the variance of the dataset in a minimal number of dimensions.

Feature selection is often discussed as a form of compression, as it reduces the data that must be processed by machine learning algorithms and models. This has several advantages, including reducing model complexity and improving performance, but selected features often have high variances and entropies that typically coincide with good predictive performance. This bias toward high entropy features limits the data compression that can be achieved with those that are selected. It may be the case, particularly with the high redundancies found with vehicle telemetry and city data, that some features with lower entropies and better compression may provide comparable predictive performances.

2.1 Data compression

Data compression aims to remove redundancy, thus making the data representation smaller, in such a way that it can be restored [14]. Characteristics of city and vehicle telemetry data, including temporal consistency, noise, and signal redundancy, all support good compression. While signal redundancy is removed in performing feature selection, noise and temporal consistency are not and must be considered using other kinds of data compression. There are two broad categories of data compression, namely lossless and lossy.

Lossless compression aims to compress the data in such a way that the uncompressed version is indistinguishable from the original [14]. Typically, lossless compression inspects the frequencies of symbols, and looks for repeating symbols or sequences of symbols in the data stream. Perhaps the most simple method of compression is runlength encoding, in which symbols are encoded along with their number of consecutive repetitions. For example, the string 'AAAABBA' can be encoded as 'A4B2A1'.

Two other notable compression algorithms are LZ77 dictionary encoding [19] and Huffman coding [9]. LZ77 uses a sliding window and searches for repeating sequences, which are encoded as the length and location of its first occurrence in the window. Huffman coding produces a variable length prefix-code defining the path to the encoded symbol in a Huffman tree. Symbols that occur with higher frequencies are located closer to the root node in the tree, and thus have shorter Huffman codes. Taken together, LZ77 and Huffman encoding make up the DEFLATE compression algorithm [3], which is the basis of the ZIP file format.

Whereas lossless compression guarantees that the decompressed stream is the same as the original, lossy compression relaxes this constraint and aims only to minimise information loss. In particular, lossy compression aims to keep information where it is important and lose information where its loss will not be noticed. In MP3 audio compression, for example, the high frequencies above the human hearing range are removed. For vehicle telemetry data, similar components of the signals can be removed if they are not useful to further analysis. Some signals such as vehicle speed, for example, contain noise that may even be detrimental to analyses.

The Discrete Wavelet Transform (DWT) can be used for compression and operates by extracting two signals that are each half the length of the original [1]. The first represents an approximation of the original signal, and is referred to as the low frequency (LF) component. The second is the high frequency (HF) component, and is a representation of the detail in the original signal. The LF and HF components are produced by a convolution of the original signal with wavelet kernels, followed by a down-sampling by a factor of 2. Typically the HF component contains many small values, and can be considered the noise component of the original signal. By quantising this component and applying a threshold, many of these small values become zero and can be encoded very efficiently using lossless compression methods such as runlength encoding. This quantisation introduces errors into the signal when it is reconstructed, but the error is minimised because only the HF component (detail) is affected. This process can be performed recursively to the LF component, further increasing the potential for lossless compression of the coefficients.

3 MRMRAC

To select features that provide good predictive performance and also have good compression, we add a compressibility factor into the feature selection process. Specifically, we introduce MRMRAC, which is an extension of the widely used MRMR feature selection framework [4, 13]. While MRMR is used as the basis for our compression-aware feature selection, the compressibility factor could be introduced into many other feature selection approaches, including wrappers and feature clustering. MRMR was chosen as a basis due to its widespread use and due to the simplicity of introducing an extra term into the selection criteria.

The MRMR framework assesses relevancy and redundancy to produce a performance measure for a feature set, which increases with higher relevance and decreases with higher redundancies [4, 13]. MRMR has several different instantiations, defined by how relevance and redundancy is assessed, as well as how they are combined [8]. One such instantiation defines the performance of a feature set as the difference between relevance and redundancy,

$$P(S, y) = Rel(S, y) - Red(S), \tag{3}$$

where $Rel(S, y)$ is the relevancy of feature set, S, to target, y, and $Red(S)$ is its redundancy.

To assess the relevance of a feature set, the individual feature relevancies, $\rho(x_i, y)$, can be aggregated,

$$Rel(S, y) = \frac{1}{|S|} \sum_{x_i \in S} \rho(x_i, y). \tag{4}$$

We do not specify the similarity measure, $\rho(\cdot)$, which may be instantiated using any correlation measure [15, 16], such as PCC, MI [18], or the Hilbert Schmidt Independence Criterion (HSIC) [5]. The redundancy of a feature set can be assessed as the mean of all pairwise feature similarities,

$$Red(S) = \frac{1}{|S-1|^2} \sum_{\substack{x_i, x_j \in X^2, \\ x_i \neq x_j}} \rho(x_i, x_j). \tag{5}$$

MRMR then aims, as in Equation 1, to find the subset of features that maximise the performance function, $P(S)$. The number of possible feature subsets is very large, and so a forward greedy search is often employed [4], as in Equation 2. As with other filter methods, the search can be stopped when a specified number of features has been selected or if the performance estimate decreases after selecting a new feature.

To extend MRMR, we introduce a third term to represent the compressibility of a feature set,

$$Com(S) = \frac{1}{|S|} \sum_{x_i \in S} \beta(x_i), \tag{6}$$

where $\beta(x_i)$ is the compressibility of the individual feature, x_i. Compressibility of an individual features can typically be assessed using entropy, but this is difficult to compute for vehicle telemetry signals, which are not independent and identically distributed. We therefore adopt a more direct approach, and measure the compression achieved by a compression algorithm on training samples,

$$\beta(x) = \frac{\text{CompressedBytes(x)}}{\text{OriginalBytes(x)}}. \tag{7}$$

This ratio is smaller for variables that are more easily compressed than those that do not compress well.

In simulations performed for this paper the compressibility is measured using either DEFLATE or DWT for all features. Different features are more suited to different compression methods, however, so it may be beneficial to employ a compression strategy targeted toward features being considered. For example, DEFLATE may be applied to some features and DWT to others, or groups of features may be considered all together in a more comprehensive compression strategy. These approaches incur higher computational costs during feature selection, however, compared to using the same compression method for all features.

To maximise the compression of selected features, this compressibility term must be minimised along with redundancy in the performance measure. In MRMRAC, the performance measure is therefore defined as,

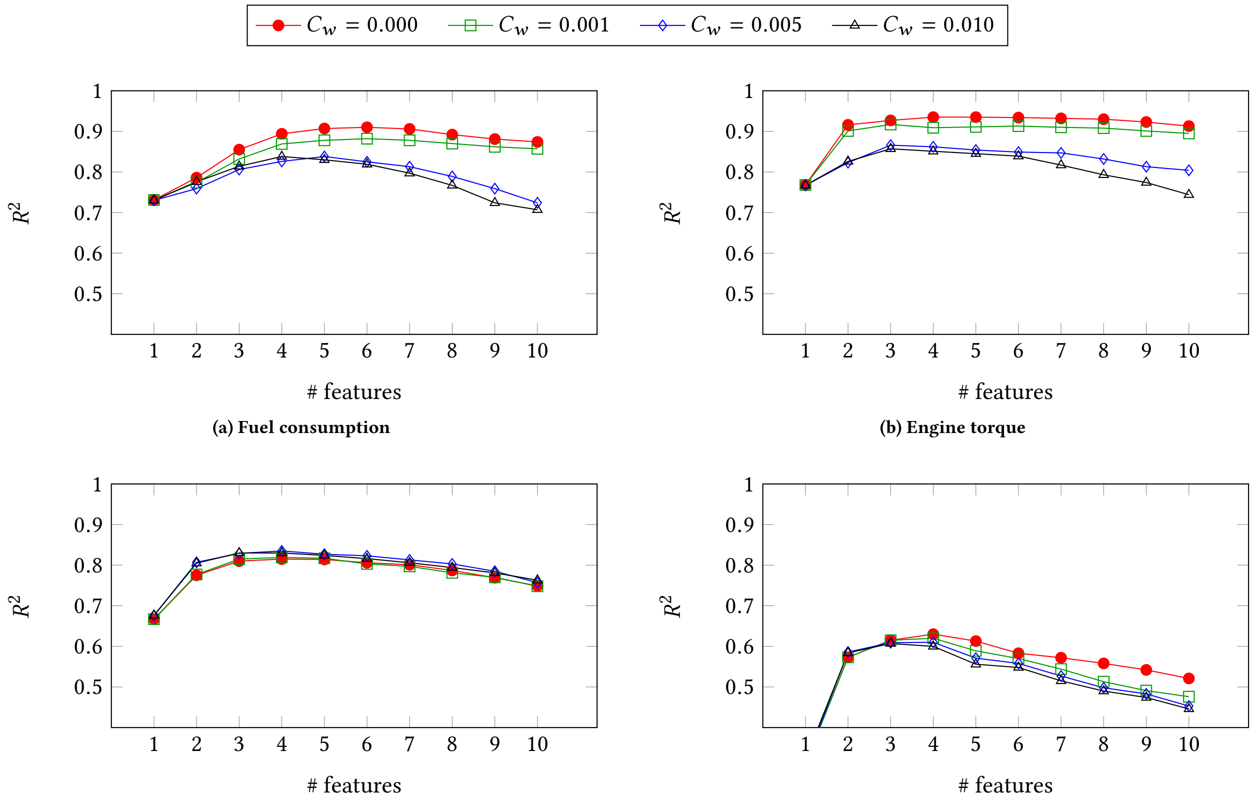
$$P(S, y) = Rel(S, y) - Red(S) - \omega_{com} \times Com(S). \tag{8}$$

The weighting parameter, ω_{com} allows the level to which compression is considered during the selection process to be varied. The smaller the value of ω_{com} the less compression is considered. In particular, a value of $\omega_{com} = 0$ means that MRMRAC is equivalent to MRMR selection.

4 RESULTS

In this section we provide results for MRMRAC feature selection as set out in Section 3. The results were obtained using the Location Extraction Dataset (LED). The LED consists of over 1900 vehicle telemetry signals collected over 72 journeys in an urban environment, while performing various pick-up and drop-off scenarios. Data was sampled at 10Hz, and the mean length of each journey was 19.7 minutes, the standard deviation of journey lengths was 8.2 minutes, and the range was 29.1 minutes. All signals with names containing the strings ‘Time’ and ‘Minutes’ were removed prior to any feature selection or model learning, as they were found to be detrimental to the results due to each sample having a unique value.

Four target variables were extracted from the data, namely the instantaneous fuel consumption, the engine torque, the vehicle speed, and the gear position. For predicting fuel consumption, all signals with names containing the strings ‘Fuel’ and ‘Torq’ were removed from the data. It was found that these signals provided



(a) Fuel consumption

(b) Engine torque

(c) Vehicle speed

(d) Gear position

Figure 1: Mean R^2 scores for different numbers of features over train-test cycles for each journey.

very high performances with only one or two features, which impedes showcasing the differences in feature selection algorithms. The other signals also had many duplicates with similar names. For predicting engine torque, therefore, all signals with names containing the string ‘Torq’ were removed. Similarly, signals containing the string ‘Speed’ were removed for predicting vehicle speed, and signals with the string ‘Gear’ were removed for predicting gear position.

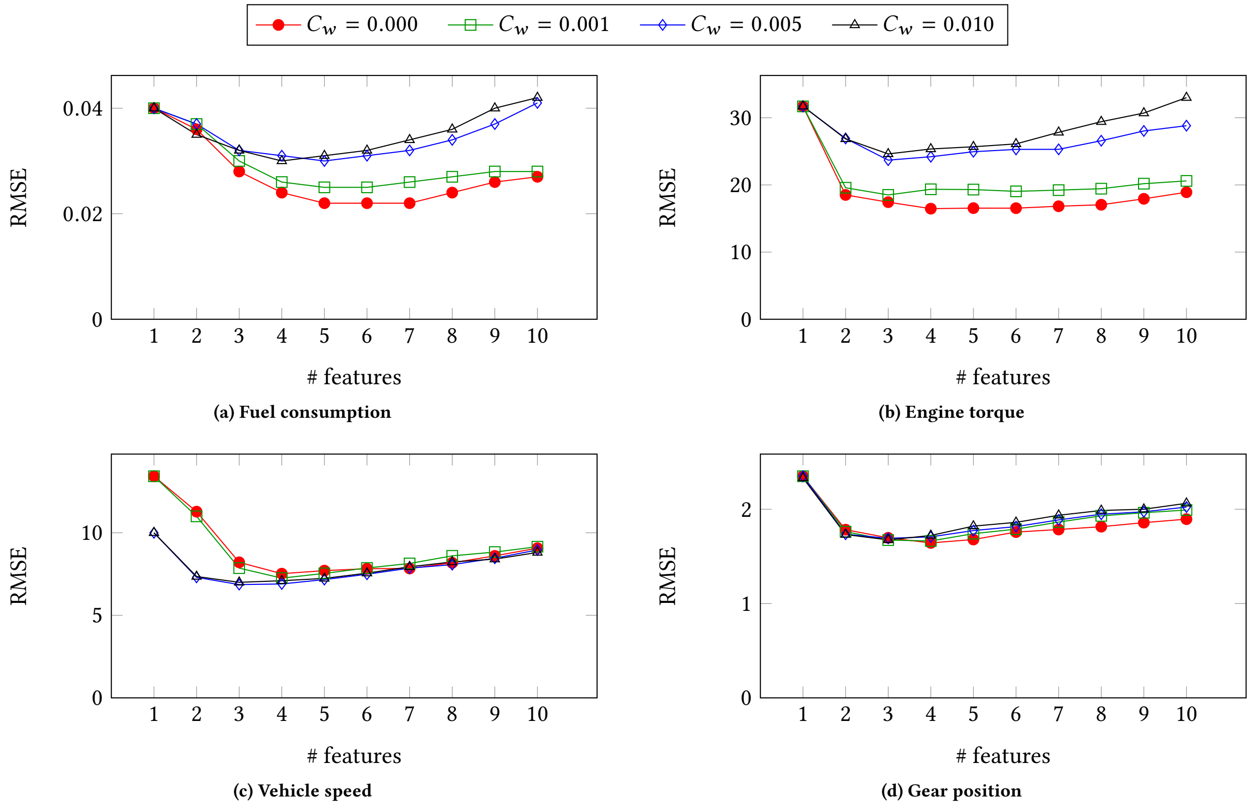
To obtain the presented results, we performed a train-test cycle on each of the 72 journey datasets. In each case, data from the first 70% of the journey in time was used as training data and the remaining data was used for testing. This provided training datasets with between 2450 and 14808 samples (mean 8282), and testing datasets with between 1050 and 3450 samples (mean 6346). In all cases there were a total of 1910 signals prior to selection. Using the training data, signals were selected from the telemetry data using MRMRAC (Equation 8) with different values for ω_{com}. The similarity measure used in selecting features was PCC in all cases. We found that PCC provided comparable performances and provided similar results to alternatives such as MI and HSIC, and was chosen due to its significantly lower computational requirements.

Once features were selected, they were used to train a Support Vector Machine (SVM) that predicted the target variable. Predictions were then made for each sample in the testing data and both R^2 and RMSE were computed to measure accuracy of the learned models. Finally, the overall compression was measured on the testing data for the selected features, using both DEFLATE (with a compression level of 6) and DWT (with coefficients compressed using DEFLATE). The DWT compression used the Haar wavelet and consisted of three levels. The results presented are the mean performances over the 72 train-test cycles.

4.1 Predictive performance

The R^2 performances when selecting signals using MRMRAC with different values of ω_{com} are shown in Figure 1. In all cases during feature selection, the DEFLATE compression algorithm was used when measuring compressibility of a signal (as in Equation 7). The highest performances for all target variables were achieved using MRMRAC with $\omega_{com} = 0$, which is equivalent to MRMR. When predicting the fuel consumption the maximum performance was achieved with around six features, whereas around four features were required for the highest R^2 performances for engine torque, vehicle speed, and gear position.

To accompany the R^2 performance, Figure 2 shows the mean RMSE scores for predicting each target variable using features selected by MRMRAC with different values of ω_{com}. In general, performances decreased with higher values of ω_{com}, due to the increased consideration of compression when assessing features. For the fuel consumption and engine torque signals there was a significant decrease in performance for $\omega_{com} = 0.005$ or $\omega_{com} =$



(a) Fuel consumption

(b) Engine torque

(c) Vehicle speed

(d) Gear position

Figure 2: Mean RMSE scores for different numbers of features over train-test cycles for each journey. Note that the y-axis in each case has a different scale due to the different ranges of the target variables.

0.01 when compared to smaller values of ω_{com}. The decreases in R^2 performances for vehicle speed and gear position were not significant, as was the case in RMSE for gear position. The RMSE for vehicle speed with one, two, or three features, was significant, but with four features the RMSE values were again similar.

4.2 Compression performance

Although the predictive performances were lower when considering compression in the selection process, the aim is to improve compression of the selected features. The plots on the left of Figure 3 (Figures 3(a), (c), (e) and (g)) show the mean numbers of bytes that were used to represent the testing data when the features were compressed using DEFLATE. For the fuel consumption, engine torque, and gear position targets, the compressed forms of features selected by MRMR (when $\omega_{com} = 0$) generally required significantly more bytes than when $\omega_{com} > 0$. Principally, this was the case for the numbers of features required for the highest predictive performances (as displayed in Figure 1). The features selected by MRMR for vehicle speed already had good compression, so the difference when considering compression was small.

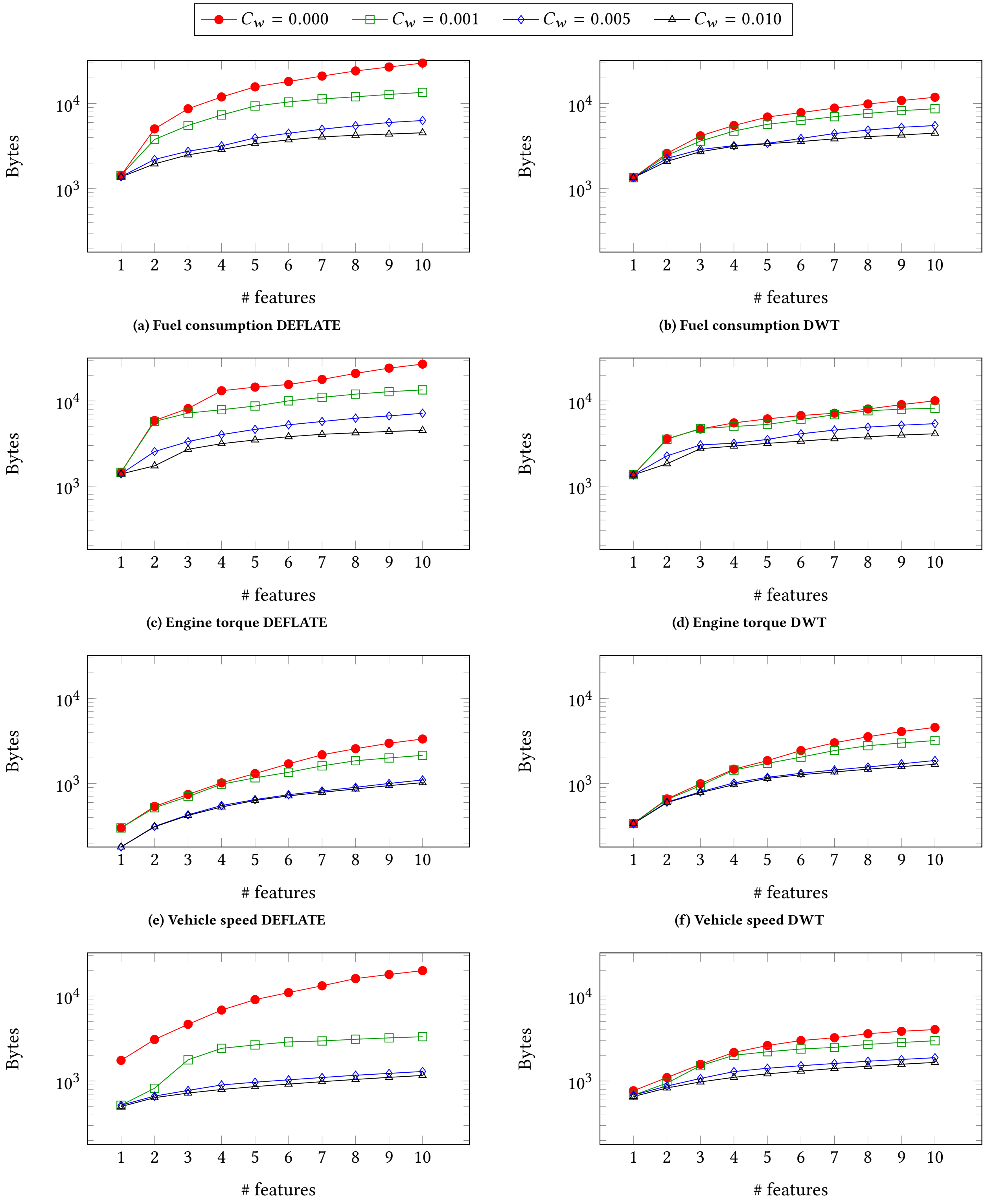
To consider only lossless compression provides an incomplete picture because the selected features may compress well, with only a small error, using lossy compression. Figures 3(b), (d), (f), and (h) show the number of bytes required to store the features when using DWT compression. In all cases, fewer bytes were required when using DWT compression than were required for DEFLATE compression. As a result, the difference in the number of bytes required by MRMR and MRMRAC was smaller. There were still more bytes required by MRMR than were required by MRMRAC, in particular for higher values of ω_{com} and when more features were selected.

4.3 DWT error

Unlike DEFLATE, which is a lossless compression method, DWT compression is lossy and introduces errors into data to which it is applied. Because multiple signals with different ranges are being analysed (i.e. the range of vehicle speed is different to steering wheel angle), the error is measured as the mean percentage error,

$$\frac{1}{|X|} \sum_{x \in X, x' \in X'} \frac{x' - x}{x}, \tag{9}$$

where X' is the signal after DWT compression and reconstruction, and X is the observed signal with $|X|$ samples. Figure 4 shows the average error of selected features introduced by using DWT compression. The largest average error in the signals was around 0.013, with the lowest errors being for features selected to predict gear position. When selecting features to predict fuel consumption, engine torque, and gear position, the error was slightly larger in general. When predicting vehicle speed, the error was slightly smaller.



(a) Fuel consumption DEFLATE

(b) Fuel consumption DWT

(c) Engine torque DEFLATE

(d) Engine torque DWT

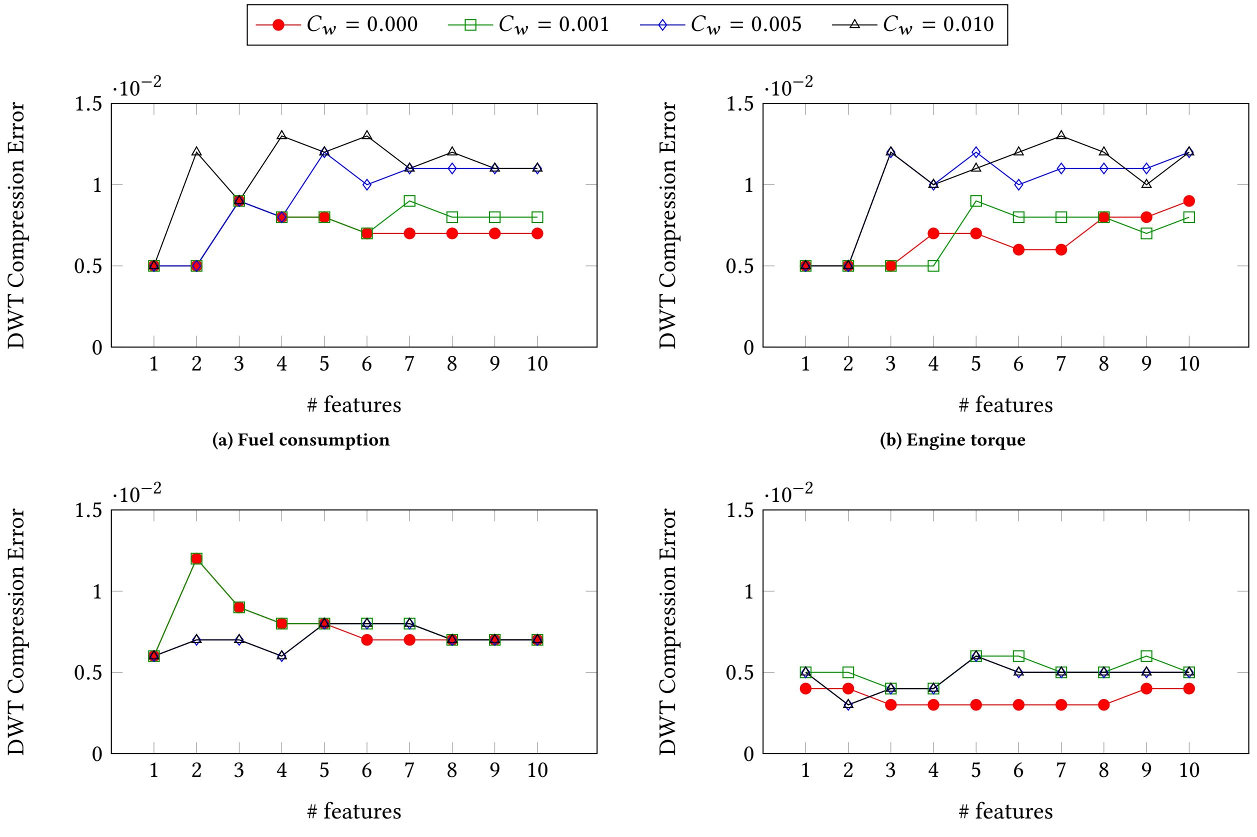
(e) Vehicle speed DEFLATE

(f) Vehicle speed DWT

(g) Gear position DEFLATE

(h) Gear position DWT

Figure 3: Mean bytes used by (left) DEFLATE and (right) DWT to represent testing samples over all train-test cycles.



(a) Fuel consumption

(b) Engine torque

(c) Vehicle speed

(d) Gear position

Figure 4: Average error (measured as percentage of actual value) for signals after reconstructed from DWT representations.

Although signal errors are an important consideration, it is their effect on model performances that is of particular interest. Table 1 shows the mean predictive and compression performances for features selected by MRMRAC. The tables on the left show results when selecting features and measuring compressibility using DEFLATE, and summarise the best performing configuration (i.e. lowest R^2) from Figures 1, 2, and 3. In these cases, the raw signal values are used at all stages of selecting, training, and testing. The tables on the right show the best performing results when selecting using DWT to measure compressibility of signals. In these results, the signals were selected and the SVMs were trained and tested using them after they were compressed and decompressed with DWT (i.e. signals had errors introduced by the lossy compression).

In general the predictive performances were slightly improved when using DWT rather than DEFLATE compression. This is likely because DWT is also a method for noise reduction, where the information loss is confined primarily to signal noise. These cleaner signals often enable models to fit more easily to them, improving performances. The number of features required for the best performances was in general very similar, and only for engine torque did the DWT compression require more features.

The DEFLATE compression of features selected was significantly worse for features selected by MRMRAC using DWT as a measure of compressibility rather than DEFLATE. This is most clearly observed when selecting features and predicting the gear position with $\omega_{com} > 0.001$. When using DEFLATE as a compressibility measure (Table 1(g)) during selection, the number of bytes required by DEFLATE to represent 4 signals was less than 2500. Using DWT as a compressibility measure (Table 1(h)) the number of bytes required for DEFLATE was over 8500. Conversely, the DWT compression was in general improved by a small margin when selecting the same number of features.

4.4 Summary

In summary, these results have demonstrated a trade-off between predictive performance of selected features and their compression. When compression was not considered in the feature selection process, as in MRMR, the performances were highest but compression of the selected features was poor. As compression was increasingly considered, by using higher values of ω_{com} in MRMRAC, performance decreased but compression improved. The difference in compression was most clear for lossless DEFLATE compression, which does not introduce any errors into the compressed features. DWT compression introduced errors into the signals, but these errors did not affect the performances of models that used them and fewer bytes were required to represent them. Having said this, MRMRAC selected features that were more easily compressed by

(a) Fuel consumption DEFLATE compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	6	0.910	0.022	18153	7831
0.001	6	0.882	0.025	10434	6320
0.005	5	0.838	0.030	3951	3402
0.010	4	0.838	0.030	2887	3147

(b) Fuel consumption DWT compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	6	0.912	0.022	18084	7839
0.001	5	0.900	0.023	14785	6334
0.005	5	0.867	0.027	10597	4244
0.010	4	0.862	0.028	8365	3302

(c) Vehicle speed DEFLATE compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	4	0.815	7.522	1020	1476
0.001	4	0.819	7.253	984	1442
0.005	4	0.835	6.900	553	1020
0.010	4	0.830	7.095	531	971

(d) Vehicle speed DWT compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	4	0.815	7.570	1020	1475
0.001	5	0.816	7.531	1146	1529
0.005	4	0.802	7.960	768	978
0.010	4	0.810	7.689	739	931

(e) Engine torque DEFLATE compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	4	0.935	16.467	13200	5536
0.001	3	0.917	18.513	7211	4750
0.005	3	0.866	23.693	3336	3049
0.010	3	0.857	24.619	2709	2754

(f) Engine torque DWT compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	6	0.939	16.422	15724	6704
0.001	5	0.937	16.624	14330	6096
0.005	5	0.927	18.097	13379	5711
0.010	4	0.906	20.243	9607	3842

(g) Gear position DEFLATE compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	4	0.630	1.643	6820	2170
0.001	4	0.620	1.664	2423	2009
0.005	4	0.610	1.707	900	1292
0.010	3	0.607	1.675	723	977

(h) Gear position DWT compressibility

ω_{com}	#	R^2	RMSE	DEFLATE (bytes)	DWT (bytes)
0.000	4	0.638	1.651	6837	2114
0.001	4	0.627	1.656	8601	1226
0.005	4	0.623	1.679	9357	1005
0.010	3	0.616	1.666	6808	796

Table 1: Mean predictive (R^2 and RMSE) and compression (bytes) performances for features selected by MRMRAC using (left) DEFLATE (right) DWT when measuring compressibility. In each case, results are shown for the number of features (#) that achieved the lowest R^2 score.

either DEFLATE or DWT than did MRMR, at the cost of slightly lower predictive performances.

5 CONCLUSION

In this paper, we have introduced the MRMRAC feature selection framework, which extends MRMR feature selection to consider the compressibility of features. Using vehicle telemetry data collected from a vehicle driven on town roads, and four different target variables, we have demonstrated MRMRAC in selecting features while considering compression to different extents. The results showed that considering compression reduced predictive performance of the selected features, but that their compression improved significantly.

As future work we intend to analyse the bounds of performance and further investigate MRMRAC using more datasets from different domains and with different characteristics. We also intend to investigate compressibility selection alongside other measures of similarity such as MI or HSIC, and evaluate the predictive performance of selected features in different kinds of model such as linear regression and Gaussian processes. Finally, we aim to introduce the the compressibility factor into other feature selection mechanisms such as wrappers.

ACKNOWLEDGEMENTS

This work was supported by Jaguar Land Rover and the UK-EPSRC grant EP/N012380/1 as part of the jointly funded Towards Autonomy: Smart and Connected Control (TASCC) Programme.

REFERENCES

[1] Paul S Addison. 2017. *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance.* CRC press.

[2] Girish Chandrashekar and Ferat Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16 – 28.

[3] P. Deutsch. 1996. *DEFLATE Compressed Data Format Specification version 1.3. RFC 1951.* Technical Report. Internet Engineering Task Force.

[4] Chris Ding and Hanchuan Peng. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology* 3, 02 (2005), 185–205.

[5] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. 2005. Measuring statistical dependence with Hilbert-Schmidt norms. In *International conference on algorithmic learning theory.* Springer, 63–77.

[6] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.

[7] John Harding, Gregory Powell, Rebecca Yoon, Joshua Fikentscher, Charlene Doyle, Dana Sade, Mike Lukuc, Jim Simons, and Jing Wang. 2014. *Vehicle-to-vehicle communications: Readiness of V2V technology for application.* Report DOT HS 812 014. National Highway Traffic Safety Administration, Washington, DC.

[8] Gunawan Herman, Bang Zhang, Yang Wang, Getian Ye, and Fang Chen. 2013. Mutual information-based method for selecting informative feature sets. *Pattern Recognition* 46, 12 (2013), 3315 – 3327.

[9] David A Huffman. 1952. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* 40, 9 (1952), 1098–1101.

[10] Jung-Yi Jiang, Ren-Jia Liou, and Shie-Jue Lee. 2011. A fuzzy self-constructing feature clustering algorithm for text classification. *IEEE transactions on knowledge and data engineering* 23, 3 (2011), 335–349.

[11] Alan Jović, Karla Brkić, and Nikola Bogunović. 2015. A review of feature selection methods with applications. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on.* IEEE, 1200–1205.

[12] Vipin Kumar and Sonajharia Minz. 2014. Feature Selection: A Literature Review. *Smart Computing Review* 4, 3 (2014), 211–229.

[13] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1226–1238.

[14] David Salomon and Giovanni Motta. 2010. *Handbook of data compression.* Springer Science & Business Media.

[15] Phillip Taylor. 2015. *Data mining of vehicle telemetry data.* Ph.D. Dissertation. University of Warwick, Coventry, UK.

[16] Phillip Taylor, Nathan Griffiths, and Abhir Bhalerao. 2015. Redundant feature selection using permutation methods. In *Automatic machine learning workshop.* 1–8.

[17] Christian Weiß. 2011. V2X communication in Europe âĂŞ From research projects towards standardization and field testing of vehicle communication technology. *Computer Networks* 55, 14 (2011), 3103 – 3119.

[18] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. 2016. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

[19] J. Ziv and A. Lempel. 1978. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory* 24, 5 (September 1978), 530–536.