Road Type Classification through Data Mining

Phillip Taylor, Sarabjot Singh Anand, Nathan Griffiths, Fatimah Adamu-Fika
{phillip.taylor, s.s.anand, nathan.griffiths, f.adamu-fika}@warwick.ac.uk
Department of Computer Science, University of Warwick, Coventry, UK

Alain Dunoyer, Thomas Popham, Xu Zhou, Adam Gelencser
{adunoyer, tpopham, xzhou1, agelencs}@jaguarlandrover.com
Jaguar Land Rover, Warwick, UK

ABSTRACT

In this paper we investigate data mining approaches to road type classification based on CAN (controller area network) bus data collected from vehicles on UK roads. We consider three related classification problems: road type (A, B, C and Motorway), signage (None, White, Green and Blue) and carriageway type (Single or Double). Knowledge of these classifications has a number of uses, including tuning the engine and adapting the user interface according to the situation. Furthermore, the current road type and surrounding area gives an indication of the driver's workload. In a residential area the driver is likely to be overloaded, while they may be under stimulated on a highway. Several data mining and temporal analysis techniques are investigated, along with selected ensemble classifiers and initial attempts to deal with a class imbalance present in the data. We find that the Random Forest ensemble algorithm has the best performance, with an AUC of 0.89 when used with a wavelet-Gaussian summary of the previous 2.5 seconds of speed and steering wheel angle recordings. We show that this technique is at least as good as a model-based solution that was manually created using domain expertise.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

Keywords

CAN bus data; road type classification; machine learning

1. INTRODUCTION

In modern road transport systems it is useful to adapt the vehicle and its user interface to the current situation or environment. For example, when a lane departure warning system (LDWS) detects an unintentional lane deviation it can alert the driver, aid the driver in correcting the deviation, or turn on the indicator light. This system improves safety on multi-lane roads, but is inconvenient in a residential setting where the driver has to move in and out of lane to avoid parked cars, bicycles, or buses. Therefore, the LDWS should be enabled on multi-lane roads, and disabled elsewhere. A further example of adaptation is to alter the engine configuration to maximize efficiency according to different road types [8].

Murphey *et al.* use features extracted from the speed profile within a time window to predict the driving environment, including road type and congestion levels [8]. Features include its standard deviation, average, maximum, the distance travelled and percentage of time spent in a certain speed or acceleration interval. The road type categorisation consists of 11 drive cycles relating to US road types such as freeway, freeway ramp, arterial, and local [11]. Using an artificial neural network (ANN) they predict the road type and congestion level, which is subsequently used for energy optimization in hybrid vehicles.

Tang and Breckon apply image processing to the road classification problem [10]. They present a method to analyse a real-time image of the road by extracting colour, texture and edge features from three subregions of the image: road, road side and road edge. Two classification problems are considered, namely: (i) on or off road, and (ii) off-road, urban, major/trunk road or multi-lane motorway/carriageway. Again an ANN is used, which is shown to have a 86% success rate on the four class problem and 97% on the two class problem. However, the authors do not consider poor light conditions, which are known to have a detrimental affect on the performance of image processing algorithms

General examples of machine learning on data collected from the CAN bus are in driver monitoring, including fatigue detection [4], driver prediction [7, 12], and driver assistance systems [6, 5]. These driver monitoring examples make use of pedal and steering wheel angle (SWA) measurements, since these are the controls with which the driver is in constant contact. These cases demonstrate the potential for inferring information about the environment and driving style from data collected from the CAN bus.

This paper investigates data mining approaches to the road classification problem. Although we use a global positioning system (GPS) sensor to generate ground truth, GPS sensors and mapping software are not present on all vehicles and require up-to-date accurate maps which are may not be available in all markets and regions. Therefore, rather than using GPS for road type classification, we use other sensors

AutomotiveUI'12, October 17-19, Portsmouth, NH, USA.

which are already installed in the vehicle and accessible via the vehicle CAN bus. Such sensors provide data including steering wheel angle, wheel speed, gear position, and suspension movement for each wheel. The CAN bus enables the communication between sensors and actuators in the vehicle via a message-based protocol, without a central host. Messages sent between devices in the vehicle can be recorded and post-processed in order to sample sensor measurements at a certain frequency.

2. DATA

Three separate classifications of UK roads are considered, namely, road type, sign type and carriageway type. Road type classifications are taken from the UK road classes: A, B, C and motorway (M). In the UK, A roads are main roads, including both trunk and residential; B roads are smaller and more local roadways; C roads are car parks, lay-bys or other unnamed roads; and motorways are highways. The sign type classification relates to the colour of signs found on the road, namely white (W), green (G) and blue (B), with a fourth class indicating no signs on the road (N). All UK motorways are blue signed, all B roads are white signed and all C roads are unsigned. A roads, however, are either white or green signed, with green signs typically used on major roads with higher speed limits, meaning that there is an overlap between the two classification problems. The third classification, carriageway type, divides roads into those with single (S) lanes or multiple/dual (D) lanes. Here, all motorways have multiple lanes, along with 20% of A roads — the majority of which are green signed. Approximately 1% of all B roads are dual lane carriageway, and likewise for white signed roads. All C roads, and therefore unsigned roads, are single lane carriageways.

The data was collected over 10 drives in the same car with multiple drivers. Data is recorded and sampled at 20Hz with 17 sensor measurements present in each sample. The mean length of the drives is 71 minutes, with the longest lasting for 115 minutes and the shortest for 50. Labels are given to samples by hand using Google Earth, into the classifications outlined above. In the resulting dataset, approximately 70% of instances which are A roads are also green signed.

In the data there is a class imbalance, for example there is a 5:1 ratio of single carriageway examples to dual carriageways, and a smaller number of motorways than other classifications, as shown in Table 1. Because of this, in this paper we consider approaches to dealing with class imbalance.

2.1 Class imbalance

Training machine learning models with class imbalanced data can adversely affect accuracy [3]. This problem is generally addressed by oversampling the minority class or undersampling the majority class. In oversampling, random replication of examples can be performed, but this introduces a bias as multiple instances in the minority class will be the same, which can lead to model overfitting. In undersampling, random deletion of examples from the majority class can be performed, meaning that some information is lost from the majority class which may be necessary for the model to capture underlying concepts.

Label	Percent (%)
Single carriageway	85
Dual carriageway	15
A road	48
B road	26
C road	21
Motorway	5
White signed	40
Green signed	34
Not Signed	21
Blue signed	5

Table 1: Label counts for the data.

More complex methods which address the problem of imbalance include Synthetic Minority Oversampling TEchnique (SMOTE) [3] and active learning [9]. In SMOTE, synthetic examples are generated along the hyper-planes between positive examples and their k-nearest neighbours. This reduces bias introduced through replication, minimising model overfitting. However, this method does increase the variance of positive examples while over-generalizing them.

Active learning aims to maximise accuracy with minimal training data. This is achieved by the learning algorithm querying an information source for the labels of selectively picked instances from a pool of unlabelled data. A base classification algorithm is first trained on a subset of the full training data. The instance which the base classifier is least confident in classifying is then added to this subset, and a new model is trained with it. This is repeated until a given number of instances are added to the training subset. With regard to addressing class imbalance, it is expected that the majority class will be undersampled as instances that do not contribute to the definition of the class boundary will not be selected.

These class imbalance methods have been shown to be effective for binary classification problems, but do not transfer to multi-class problems. In future work we will consider alternative techniques for handling multi-class problems with skewed distributions.

2.2 Dimensionality Reduction

Both attribute selection and principle components analysis (PCA) were investigated for reducing the dimensionality of the data. The information gain for dataset S and attribute A is defined as:

$$IG(S,A) = H(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} H(S_v)$$

where S_v is the partition of the data S defined by the value v of attribute A, and $H(S)$ is the entropy of set S. The attributes with the 10 highest information gains are shown in Table 2. The attributes most related to the class attribute in all classifications are vehicle speed, SWA and gear position. After these, the information gain is substantially lower.

Because of the large drop in information gain after the top three attributes, these three attributes are the natural candidates for selection. However, gear position is closely related

Attribute	Road	Sign	Carriageway
Vehicle speed	0.520	0.519	0.193
Gear position	0.406	0.387	0.075
SWA	0.201	0.214	0.049
Lateral acceleration	0.067	0.066	0.009
SWA speed	0.059	0.062	0.017
Front-left suspension	0.039	0.043	0.011
Is brake pressed	0.038	0.039	0.008
Front-right suspension	0.036	0.041	0.017
Rear-left suspension	0.033	0.035	0.014
Rear-right suspension	0.031	0.033	0.009

Table 2: Information gain of top 10 attributes.

to vehicle speed, as different gears are required for different speeds. Furthermore, gearboxes in different vehicles have different ratios, and so gear position is not directly transferable between vehicles. Therefore, to produce models that are as simple and transferable as possible, this attribute is not considered. Initial tests with the J48 algorithm show that the AUC is only marginally affected by omitting gear position, decreasing by 0.03 in both the road and sign classifications and not at all for carriageway classification.

Using a number of correlated input attributes can reduce the accuracy of the models learnt. Hence a frequently used pre-processing step is to transform the instance space using a linear transformation into a lower dimensional, uncorrelated space. One such method is PCA, in which Eigen vectors of the covariance matrix define the new projected dimensions (called principal components). The corresponding Eigen values provide a measure of the extent to which the variance within the data is captured within this dimension. In noisy domains such as ours we would expect a large percentage of the variance to be captured by a small number of dimensions. Figure 1 shows that almost 80% of the variance in our data is captured by the first five principal components.

2.3 Temporal summary of the data

As the data collected from the CAN bus is a temporal sequence, current measurements will be dependent upon past measurements. This means consecutive readings from sensors are likely to be very similar, especially given the relatively high frequency of 20Hz. For example, a velocity reading of 70kph at one instant means the sensor is unlikely to read 0kph in the next. Therefore, it makes sense to analyse the data in time windows.

The data is first split into time windows of size δ. Then, each window is summarised through some transform. This means, the instance I_t at time t, becomes a summary of instances $I_{t-\delta:t}$. The true classification given to a window is that of the instance I_t, i.e., the final instant in the window. In this paper we consider the following summary methods: Gaussian distribution (G), discrete Fourier transform (DFT) and discrete wavelet transform (DWT).

Gaussian distribution

This method assumes that observed signals contain Gaussian noise. It aims to estimate the signal's true value through its mean, μ, and standard deviation, σ. The instance $I_{t-\delta:t}$

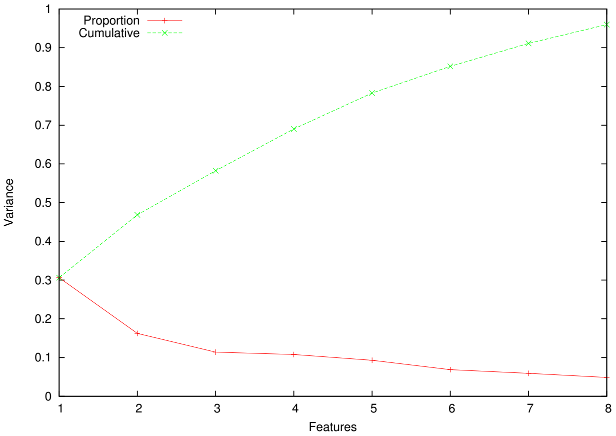

Figure 1: Variance (y-axis) of principle components by number of features (x-axis).

becomes:

$$\{Velocity(\mu)_{t-\delta:t}; Velocity(\sigma)_{t-\delta:t};\\ SWA(\mu)_{t-\delta:t}; SWA(\sigma)_{t-\delta:t}\}$$

Since each signal is represented by two values, the dimensionality of the problem is doubled.

DFT

The DFT of a sequence x is defined as:

$$DFT(x)_k = \sum_{n=0}^{|x|-1} x_n \cdot e^{-i2\pi\frac{k}{|x|}n}$$

where $k = \{0, \ldots, |x| - 1\}$ and $DFT(x)_k$ are the complex coefficients of the sinusoidal components of x. It is intuitive to compute the magnitude of these components, producing a power spectrum of component frequencies of the original signal.

$$|DFT(x)_k| = \sqrt{real(DFT(x)_k)^2 + imag(DFT(x)_k)^2}$$

In practice it is not necessary to use all these coefficients, as many after the first few have magnitudes of 0. In our data sets we observed that the performance of the classification algorithm does not degrade much when using just the first 5 coefficients. This can be seen in Figure 2, which shows AUCs for the J48 classification algorithm estimated using 10-fold cross validation using various numbers of coefficients. Therefore, for the classification problems the first 5 coefficients are chosen, producing the instance $I_{t-\delta:t}$ of:

$$\{|DFT(Velocity_{t-\delta:t})_{1:5}|; |DFT(SWA_{t-\delta:t})_{1:5}|\}$$

DWT

A one level DWT is performed, producing a set of approximation and detail coefficients. These coefficients are computed by passing the signal through low pass filter and high pass filters, with impulse responses of g and h, respectively.

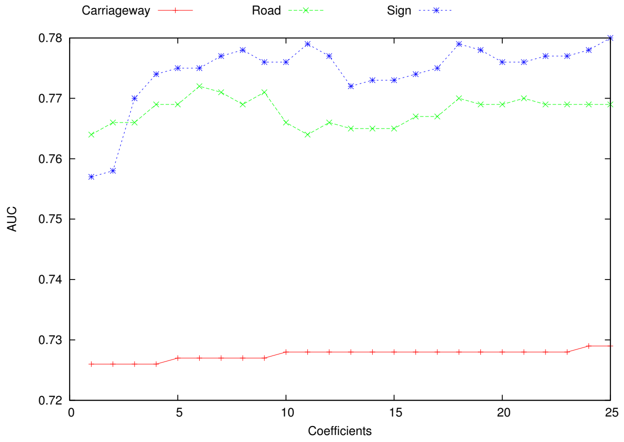

Figure 2: The AUC for a J48 classifier (y-axis) varying the number of Fourier coefficients (x-axis).

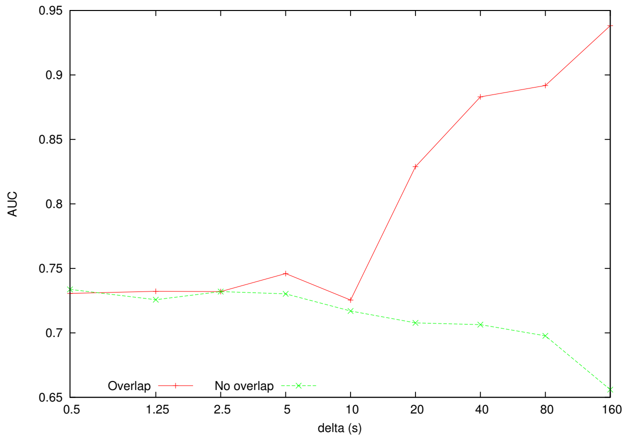

Figure 3: The AUC for carriageway classification using J48 (y-axis) with varying window sizes in seconds (x-axis) for the Overlap and No Overlap windowing methods.

Both are then sub-sampled by 2, producing two outputs, half the length of the original signal.

$$Approximation = (x * g) \downarrow 2; Detail = (x * h) \downarrow 2$$

The Haar wavelet, $[1, -1, 0]$, is chosen for use in the transform due to its simplicity. The high pass filter here sums each pair of values in the sequence, whereas the low pass filter takes their differences. This gives $g = [1, 1]$ and $h = [1, -1]$.

After performing the DWT on the time window, it is then possible to perform a second temporal summary method, such as the DFT, on both the approximation and detail coefficients. If the Gaussian method is to be used in conjunction with the DWT, the dimensionality of the dataset will be increased by 4 times. We refer to these as wavelet-Gaussian (DWT-G) and wavelet-Fourier (DWT-DFT) summaries respectively.

2.4 Window size

We consider two windowing methods for various window sizes, δ. First, we use a constant gap of 50 data points (i.e. 2.5s) between the initial data points of the windows that define consecutive instances. For $\delta > 2.5$s this means that consecutive instances will overlap, and so we refer to this method as Overlap. Second, we consider a gap between the initial data points in consecutive windows of 2δ, which we refer to as No Overlap. We evaluate these methods using the J48 classifier [13] with a Gaussian summary, With Overlap we find that a window larger than 10s lowers the success rate and corresponding AUC, as shown in Figure 3. This is because the overlap causes consecutive instances to be very similar, meaning that a randomized test set is likely to contain similar instances to the training data. Because of this, an evaluation bias is introduced, with evaluations showing optimistic performance for models. The No Overlap approach avoids this issue and has degrading performance with $\delta > 2.5$s. In the remainder of this paper we use No Overlap with $\delta = 2.5$s. Note that similar results are given by the Fourier summary but we the omit results for space reasons.

The steering wheel angle and vehicle speed signals are windowed using the No Overlap method and summarized using the Gaussian, DFT, DWT-G and DWT-DFT transform methods. A window size of $\delta = 2.5$s is used meaning that a gap of 5s is introduced between the start of consecutive windows. Processing the raw CAN bus data in this way gives a dataset of 8618 instances which are used to build the classification models.

3. EVALUATION RESULTS

3.1 Evaluation methodology

It is expected that data mining approaches can at least equal the performance of a model created manually by an expert using domain knowledge. We also hypothesise that approaches to dealing with the class imbalance will improve the performance of the models. Our aim is for the models built to be transferable to different vehicles, environments, weather conditions and roads.

Data from all of the 10 drives are combined, after individually undergoing the windowing process described above. A 10-fold cross validation is then used to evaluate models. When building the model on training data, a second 10-fold cross validation is performed to find the best parameters for the model, listed in Table 3. The optimal parameters are those that produce the largest area under the ROC (receiver operator characteristic) curve (AUC) after the 10-fold cross validation on the training data. These optimised parameters are then used when building the model on the full training data. This method allows for an unbiased estimate of model performance.

After this evaluation process is complete, a ROC curve and

Algorithm	Options
J48	Pruning parameters
OR	Number of discretization bins
RIPPER	Number of folds for error pruning and minimal weight of instances within split
NB	Kernel or supervised discretization
ANN	Learning rate and momentum
HMM	Number of hidden states
RF	Number of trees, their depth and number of features
Adaboost	Base classifier to use
Sampling	Amount of over or under sampling

Table 3: Classification algorithms used and their parameters to be optimised during evaluation.

its subsequent AUC value are calculated. An alternative metric for comparing classification algorithms is the success rate i.e. the number of correct classifications. We provide the 95% confidence interval on our estimate of success rate.

This evaluation process is performed using the WEKA machine learning environment [13] and the probabilistic modelling toolkit (PMTK) for Matlab[1]. Numerous classification algorithms are investigated, including decision trees (J48 which is the implementation of C4.5 in WEKA), Naïve Bayes (NB) and ANNs. In the case of ANNs, initial experiments suggested that the highest AUC values are produced with one hidden layer, containing 5 nodes in the two class problem and 7 nodes in the four class problem. Two decision rule algorithms are also investigated, the simple one rule (OR) algorithm and the more complex Repeated Incremental Pruning to Produce Error Reduction (RIPPER) algorithm (referred to as RIP in the results tables) [13]. In addition to these base classifiers two ensemble classifiers were also investigated, namely Adaboost [2] and Random Forest (RF) [13]. Base models used with the Adaboost ensemble classifier in this paper are those introduced above and are referred to by assigning the prefix A-.

We also investigate hidden Markov models (HMMs), which are trained through unsupervised learning. The most likely sequence of states to produce a sequence of observed values can be computed through the Viterbi algorithm. In our context, the hidden states relate to the road type and each classification may be split across multiple states. As HMMs model a sequence of states, the raw sequence of data from a full drive is used rather than using a temporal summary of the data. A leave one out strategy is used, where the HMM is built with data from 9 of 10 the drives in the dataset, and tested on the other. Again, the AUC and 95% confidence interval of the success rate is calculated.

3.2 Results

CAN bus data consists of a set of observations collected over time, that are influenced by the (unobserved) road type that the vehicle is travelling on, and so it provides an ideal application for HMMs. The parameters of the HMMs can be learnt from the observed sequence for a given number of hidden states. As there is heterogeneity within the set of roads assigned the same class label we learnt different HMMs with between 2 and 6 hidden states. In the carriageway type classification, the most accurate HMM was that with 6 hidden states. For both the four class problems, road and sign classification, a HMM with 5 hidden states performed best. As the HMM takes a full dataset as an input sequence, the temporal summary methods were not applied, and so only results for the raw sequence are presented in Table 4. As stated previously, it is unlikely that the road type will change very rapidly. This fact is reflected in the transition matrix produced by the training phase of the HMMs, which is always very diagonal. The probabilities of changing states are also highest between A and B roads, or white and green signed roads. This is most likely because of the similarity of these roads in driving experience and style.

	Hidden States	AUC	Success rate
Carriageway	6	0.707	0.89±.007
Road	5	0.568	0.544±.011
Sign	5	0.587	0.592±.01

Table 4: Number of hidden states, AUC, and 95% confidence interval of success rate for HMMs.

Clear differences between the three classification problems can be seen in their success rate metrics. The carriageway type problem, a two class problem, is much simpler than the other four class problems. This leads to a success rate difference of around 30% between the best classifiers for the respective problems. The success rates for both the road type and sign type problems are very similar however, which reflects their similar definitions as explained previously.

Given the HMM results, we next consider whether supervised methods for learning can improve on the accuracy of the models. Here the temporal summarization methods are applied using a window size of $\delta = 2.5$s and the No Overlap approach as these were optimal settings as shown in Figure 3. The first objective was to evaluate whether attribute selection or feature extraction (PCA) would provide more accurate models within this domain. Figure 4 shows the results obtained using J48 as the number of principal components used was increased from 1 to 5, with and without using some method of handling the class imbalance. These are compared with J48 models learnt using just the SWA and Velocity attributes and Gaussian summarization. As we can see, using 2 principal components results in a significantly better model than when using a subset of features. However, when we use methods for handling the class imbalance, the feature subset outperforms the models built using principal components. Similar patterns were observed for other learning algorithms and temporal summarization approaches. Due to lack of space we are unable to present all these results here and hence focus on results obtained using the feature subset selected.

Tables 5 and 6 show the results obtained for each of the temporal summary and classification methods without considering the class imbalance. In all the classification problems, Random Forest is the most successful in terms of both AUC and success rate when the model has an evenly distributed decision threshold. Also, in each of these a Gaussian sum-

[1] PMTK — http://code.google.com/p/pmtk3

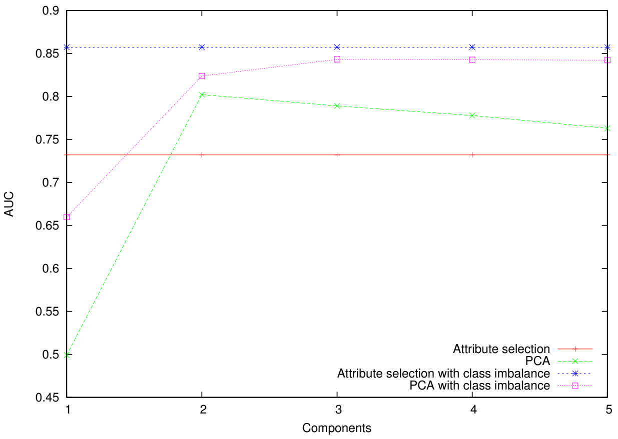

Figure 4: AUC for principal components and selected attributes using J48.

mary of the wavelet transform performs the best, albeit by a very small amount. The ROC curves for Random Forest and the DWT-G summary method for the carriageway, road and sign classification problems are shown in Figures 5, 6 and 7 respectively. The Fourier transform of the data generally performs the worst.

One surprising result in these experiments is the performance of relatively simple methods. For example, OR performs only slightly worse than the more complex decision tree, J48, and better than the multilayer perceptron in many experiments. Also, the Gaussian summary is almost as good as the other, more complex transforms. This is reflected in the rules and decision trees produced, which usually split on vehicle speed first, classifying any instances under 25kph as a single carriageway, and those over around 65kph as a motorway (and therefore also as dual carriageway).

One reason for these simple models performing well is that speeds on C roads are usually low, and they are all single carriageways. Likewise motorways are all dual carriageways and have higher speeds than other road types. The most confusion in any model's predictions are of A and B roads, which is expected as they are most similar to drive on and have similar speed limits. This can be seen in the confusion matrices produced by testing and the ROC curves, which show significantly worse performance for those classes. This means that the SWA is used in the models to differentiate between these. Similarly, in the sign classification problem, most of the confusion is seen between the white and green signed roads.

In order to address the class imbalance present in the data, under-sampling, over-sampling, SMOTE and active learning were investigated. The same evaluation procedure is used as before, with the amount of sampling as a parameter to be optimised. Results for the classification algorithms with highest AUCs are shown in Table 7. The Random Forest algorithm performs best for each of under-, over- and

	DFT	G	DWT-DFT	DWT-G
OR	0.684	0.703	0.698	0.721
NB	0.807	0.786	0.807	0.756
J48	0.727	0.732	0.732	0.736
RF	0.863	0.884	0.872	0.887
RIPPER	0.712	0.717	0.704	0.721
ANN	0.809	0.803	0.831	0.817
A-RIPPER	0.836	0.859	0.832	0.847

(a)

	DFT	G	DWT-DFT	DWT-G
OR	0.67	0.669	0.672	0.667
NB	0.67	0.685	0.668	0.639
J48	0.768	0.773	0.772	0.796
RF	0.807	0.817	0.812	0.824
ANN	0.752	0.741	0.76	0.749
RIPPER	0.748	0.75	0.751	0.759
A-J48	0.779	0.795	0.779	0.799

(b)

	DFT	G	DWT-DFT	DWT-G
OR	0.685	0.682	0.684	0.684
NB	0.712	0.699	0.66	0.621
J48	0.77	0.78	0.779	0.778
RF	0.811	0.818	0.814	0.822
RIPPER	0.759	0.768	0.764	0.768
ANN	0.754	0.749	0.763	0.755
A-J48	0.789	0.796	0.786	0.791

(c)

Table 5: AUC for (a) carriageway, (b) road, and (c) sign classification without sampling.

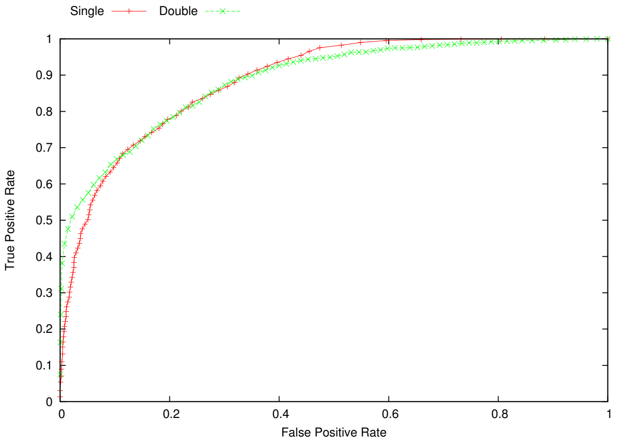

Figure 5: ROC curve for Random Forest on carriageway classification. False positive rates are shown on the x-axis, true positive rates on the y-axis.

	DFT	G	DWT-DFT	DWT-G
OR	0.887±.007	0.887±.007	0.889±.007	0.889±.007
NB	0.406±.01	0.453±.011	0.264±.009	0.267±.009
J48	0.904±.006	0.906±.006	0.906±.006	0.907±.006
RF	0.905±.006	0.91±.006	0.909±.006	0.911±.006
ANN	0.901±.006	0.908±.006	0.903±.006	0.906±.006
RIP	0.894±.006	0.889±.007	0.895±.006	0.886±.007
A-RIP	0.894±.006	0.889±.007	0.895±.006	0.885±.007

(a)

	DFT	G	DWT-DFT	DWT-G
OR	0.586±.01	0.589±.01	0.588±.01	0.586±.01
NB	0.428±.01	0.421±.01	0.353±.01	0.337±.01
J48	0.63±.01	0.645±.01	0.641±.01	0.66±.01
RF	0.648±.01	0.655±.01	0.656±.01	0.669±.01
ANN	0.633±.01	0.639±.01	0.635±.01	0.647±.01
RIP	0.585±.01	0.578±.01	0.597±.01	0.591±.01
A-J48	0.631±.01	0.644±.01	0.634±.01	0.654±.01

(b)

	DFT	G	DWT-DFT	DWT-G
OR	0.576±.01	0.575±.01	0.577±.01	0.576±.01
NB	0.495±.011	0.447±.01	0.288±.01	0.2±.008
J48	0.613±.01	0.613±.01	0.611±.01	0.617±.01
RF	0.629±.01	0.633±.01	0.631±.01	0.639±.01
ANN	0.605±.01	0.619±.01	0.608±.01	0.618±.01
RIP	0.574±.01	0.568±.01	0.581±.01	0.572±.01
A-J48	0.62±.01	0.621±.01	0.62±.01	0.62±.01

(c)

Table 6: Success rate as a 95% confidence interval for (a) carriageway, (b) road, and (c) sign classification without sampling.

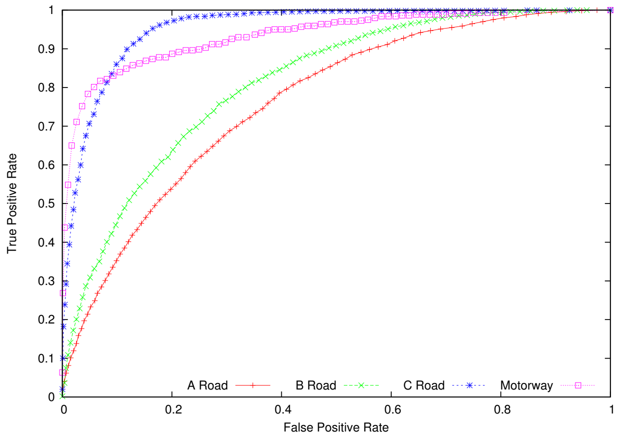

Figure 6: ROC curve for Random Forest on road classification.

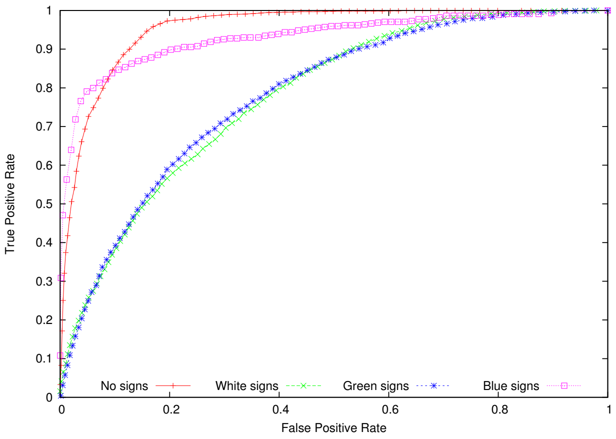

Figure 7: ROC curve for Random Forest on sign classification.

	DFT	G	DWT-DFT	DWT-G
Under-sample	0.856	0.883	0.868	0.880
Over-sample	0.849	0.881	0.863	0.882
SMOTE	0.848	0.879	0.858	0.881
Active learning	0.825	0.841	0.831	0.836

Table 7: AUC for under-sampling, over-sampling, SMOTE using Random Forest and active learning using a J48 classifier on carriageway classification.

SMOTE sampling, with best AUCs of 0.883, 0.882 and 0.881 respectively. For active learning the only base classification algorithm tested is J48, having a best AUC of 0.841.

Again in these results, it is clear the best pre-processing techniques are the Gaussian or wavelet-Gaussian summaries, with little difference between the two. Also, it is noteworthy that the simpler sampling strategies perform better than the more complex SMOTE and active learning, albeit again by a small margin. Active learning performs worst, which may be a result of its use of J48 as a base classification algorithm. This may be improved upon if other learning algorithms are used as base classifiers in the active learning process.

These approaches to dealing with class imbalance do not improve upon the AUCs found when using Random Forest without any sampling. However, for other models such as J48 and OR, a substantial improvement is seen. For example, the AUC for the J48 classifier without sampling and the wavelet-Gaussian summary, is improved by 16% to 0.854 when SMOTE sampling is used. This value is much closer to that achieved by the Random Forest algorithm, suggesting the robustness of ensemble models to class imbalance.

3.3 Comparison with model based solution

A model-based solution has also been investigated previously by a Jaguar Land Rover employee for carriageway type classification, the details of which cannot be disclosed due to commercial confidentiality. As the model-based solution does not require any training, data from each of the drives are classified and evaluated, producing an AUC and success rate. In order to compare with the results from the data mining approach, we use the same leave one out strategy as applied to HMMs to evaluate the Random Forest classifier. The mean AUC and 95% confidence intervals of the success rates of these drives are then calculated for both the model-based and data mining approaches.

The model-based solution has a mean AUC of 0.751, and a success rate 95% confidence interval of 0.909 ± .001. The Random Forest model improves substantially on this AUC, scoring 0.867. The 95% confidence interval of the success rate for the Random Forest classifier is 0.903 ± .001, which is very similar to that of the model-based solution.

4. CONCLUSION

Several data mining and temporal analysis techniques have been investigated for road type classification. UK roads are given three different classifications, road type (A, B, C and Motorway), signage (None, White, Green and Blue) and carriageway type (Single or Double). In each of these problems the Random Forest algorithm performs best when used in conjunction with a wavelet-Gaussian summary of the data. This technique has AUCs of 0.824, 0.822 and 0.887 and success rates of 0.669 ± .01, 0.639 ± .01 and 0.911 ± .006 at a 95% confidence interval for the road type, signage and carriageway classifications respectively.

Approaches to dealing with class imbalance in the carriageway problem were investigated, namely under-sampling, oversampling, SMOTE and active learning. The AUC of 0.887 given by Random Forest on un-sampled data is not improved upon by any of these techniques. However, the AUC of other classification algorithms are substantially improved upon by these techniques, by over 10% in general. The AUCs for over-, under- and SMOTE sampling are very similar, with under-sampling being the most successful by a very small margin. All three of these methods have better performance than that of active learning.

Models trained on data used in this paper have also been tested on a small dataset recorded from a different vehicle. This smaller dataset has road and carriageway labels, but not sign type. First, Random Forest with Gaussian summary is used with the same evaluation procedure as previously to gain a benchmark. Second, the classification algorithm is trained on the data used in this paper with the parameters optimised. This built model is then tested on the smaller dataset. The AUCs from these models drop by a small amount for the carriageway classification, from 0.846 to 0.799. For the road classification a larger drop is seen, from a benchmark of 0.896 to 0.813. This drop in performance may be due to slight inconsistencies in labelling the datasets (as they were done by different people) or some other factor which was not controlled. This result does show however that models built on data from one vehicle can be used, to reasonably good effect, with other vehicles.

Further, this data mining technique is shown to be at least as good as the manually created model-based solution. The AUC of the data mining approach is shown to be significantly higher than that of the model-based solution, with a very similar success rate. This gives credence to data mining being applied in this domain, over creating models by hand.

The techniques for class imbalance considered in this paper are only suitable for two class problems, meaning that the road and sign type problems have not been investigated with regards to class imbalance. In future work we will investigate methods for dealing with multi-class class imbalance, such as Error-Correcting Output Coding (ECOC) [1] in which a unique code is given to each class and a classifier is built to predict each of the digits of the code. ECOC has been successfully applied in other domains such as text classification.

5. ACKNOWLEDGEMENTS

The authors would like to thank Jaguar Land Rover and EPSRC for funding this work.

6. REFERENCES

[1] Adam Berger. Error-correcting output coding for text classification. In *IJCAI'99: Workshop on machine learning for information filtering*, 1999.

[2] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proc. of the 13th Int. Conf. on Machine Learning*, pages 148–156, 1996.

[3] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, sep 2009.

[4] Qichang He, Wei Li, and Xiumin Fan. Estimation of driver's fatigue based on steering wheel angle. In Don Harris, editor, *Engineering Psychology and Cognitive Ergonomics*, pages 145–155. Springer, 2011.

[5] J. C. McCall and M. M. Trivedi. Human behavior based predictive brake assistance. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 8–12, 0-0 2006.

[6] J. C. McCall and M. M. Trivedi. Driver behavior and situation aware brake assistance for intelligent vehicles. *Proc. of the IEEE*, 95(2):374–387, feb. 2007.

[7] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao. Lane change intent analysis using robust operators and sparse bayesian learning. *IEEE Trans. on Intelligent Transportation Systems*, 8(3):431–440, 2007.

[8] Y. L. Murphey, Zhi Hang Chen, L. Kiliaris, Jungme Park, Ming Kuang, A. Masrur, and A. Phillips. Neural learning of driving environment prediction for vehicle power management. In *IEEE Int. Joint Conf. on Neural Networks*, pages 3755–3761, 2008.

[9] Burr Settles. Active learning literature survey. Technical Report 1648, 2009.

[10] I. Tang and T. P. Breckon. Automatic road environment classification. *IEEE Trans. on Intelligent Transportation Systems*, 12(2):476–484, 2011.

[11] Thomas C. Austin Thomas R. Carlson. Development of speed correction cycles. Technical report, Sierra Research, Inc, 04 1997.

[12] C. Tran, A. Doshi, and M. M. Trivedi. Pedal error prediction by driver foot gesture analysis: A vision-based inquiry. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 577–582, 2011.

[13] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2005.