Identifying Distracted and Drowsy Drivers Using Naturalistic Driving Data

Sujay Yadawadkar, Brian Mayer, Sanket Lokegaonkar,
Mohammed Raihanul Islam, Naren Ramakrishnan
Discovery Analytics Center
Virginia Polytechnic Institute and State University
Blacksburg, VA, United States
Email: sujayr91@vt.edu, mayer2@vt.edu, sloke@vt.edu,
raihan8@vt.edu, naren@vt.edu

Miao Song, Michael Mollenhauer
Virginia Tech Transportation Institute
Virginia Polytechnic Institute and State University
Blacksburg, VA, United States
Email: msong@vtti.vt.edu, mmollen@vt.edu

***Abstract*—Driver fatigue and distraction remain significant safety issues for drivers. Despite substantial developments in driver state detection technology, a reliable system has yet to emerge. Existing systems tend to suffer from reliance on a single metric such as PERCLOS estimated from single expensive in-vehicle cameras and/or a poorly designed and tuned algorithm resulting in lack of effectiveness (high false positive rates). It is not likely that any single, real-time measure of driver drowsiness will be obtainable all of the time from the entire driver population. Therefore, a multi-variable algorithm based on sensors/variables that can be reliably obtained in real time on modern vehicles is essential. In this work, several algorithms for multivariate time-series analysis are tested on the Second Strategic Highway Research Program (SHRP2) Naturalistic Driving Study (NDS) dataset, including a statistical feature extraction method, deep learning-based long short-term memory, and video classification using convolutional neural networks. Given the amount of training and test data currently available, traditional statistical feature extraction methods outperformed the deep learning methods tested.**

***Keywords*-RNN; LSTM; SMOTE; CNN; Naturalistic Driving; Driver Monitoring**

I. Introduction

Building a robust driver monitoring system has been a challenge for many years. Although there have been significant improvements in the inference methods with the latest deep learning, machine learning techniques, estimating driver state has been a challenge mainly because of the unavailability of data collected under naturalistic driving conditions.

The Second Strategic Highway Research Program (SHRP2) Naturalistic Driving Study (NDS) contains rich data collected over 3 years where approximately 3,400 drivers participated in the study across the United States. The collection of multiple time-series sensor data and accompanying video represents the equivalent of four millennia of driving time where around 36,000 crash, near-crash, and baseline events were identified. This includes around 580 events flagged as driving under drowsy conditions and 1200 crash/near-crashes (C/NCs) of distracted drivers [1].

Building a reliable machine learning algorithm involves robust feature selection, sensor fusion, and handling imbalanced datasets. The major contributions of this work are as follows:

1) Handling the imbalance in the dataset. We consider oversampling techniques, namely SMOTE to best utilize the total available dataset.
2) Understanding important time-series precursors for driver state prediction.
3) Analyzing several methods for extracting robust cues for accurate prediction of driver state. In this regard we consider methods based on statistical feature extraction, contemporary deep learning-based approaches such as Long short-term memory (LSTM) for multivariate time-series classification and deep convolutional neural networks (CNNs) for video classification.

II. Related Work

In recent years there has been considerable research in driver monitoring systems. There has been special focus on prediction using video data by estimating features such as gaze [2], [3], 3d head pose [4], [5], eye closure rate [6], [7] etc. Although these remain highly discriminative features for driver distraction/drowsy classification, understanding the boundary conditions for classification is hard as quantifying the thresholds require labeled examples for segments of video. In addition to driver monitoring, there has also been extensive analysis of using time-series data in an automotive setting to predict a driver's future maneuver or action based on video data [8], [9], [10].

Previous work on this project [11] utilized time-series sensor features such as lane distance, accelerometer, and several other vehicle dynamic sensor measurements to develop a hand crafted drowsy state detection algorithm based on a boosting method thereby emphasizing the importance of discriminative features present in non-video time-series data. This has given better prospects for further research in understanding drowsy cues from a time-series dataset given the fact that video data collection is extremely cumbersome and involves privacy and labeling issues.

Alongside these, there has been decades of research in analyzing time-series for classification and prediction. Among these methods, time-series prediction based on statistical features, Hidden Markov Models, Autoregressive/ARIMA models have

been extensively studied [12], [13], [14], [15], [16], [17]. Recently, time-series classification using deep learning techniques, such as with LSTMs, has been researched and has shown promising results [18], [19], [20], [21].

LSTMs have been extremely popular and good for sequential prediction tasks. Recently LSTMs have led to ground breaking results in machine to machine translation, image to text translation, etc. Improved training procedure and lesser susceptibility to the vanishing gradient problem for large sequence lengths have been the keys for LSTM's success.

In this work we consider several algorithms for multivariate time-series analysis on the SHRP2 NDS dataset for driver state prediction. First, we consider analysis with time-series data without using video features. We give special focus for handling imbalance in the dataset by considering oversampling methods and understanding important discriminative features. After this, we do analysis with the latest contemporary methods, such as with LSTM with attention mechanism for time-series classification. Finally we attempt video classification using CNNs.

A. Research Impact

This research uses well researched and tested machine and deep learning techniques to identify sensors, variables, and methods that are effective in classifying drowsy/distracted drivers. The research lists the shortcomings of current techniques when applied to solving this problem. The results of this paper will help to drive future research in areas that can eventually enable application of these techniques to accomplish known problems that are impactful to society.

III. Data

A. Data Set

Data was collected exclusively from the SHRP2 NDS database. As the largest naturalistic driving dataset available worldwide, the SHRP2 NDS database offers detailed and accurate pre-crash information not available from other crash databases. This pre-crash information serves as strong and powerful evidence identifying the progression of critical driving behaviors, in addition to, traffic and vehicle dynamics. These were either captured by an installed on-board Data Acquisition System (DAS) or manually processed post-hoc by viewing video. The DAS includes forward radar; four video cameras, including one forward-facing, color, wide-angle view; accelerometers; vehicle network information; Geographic Positioning System; on-board computer vision lane tracking, plus other computer vision algorithms; and data storage capability. [22]

Data was initially collected on the vehicle and then downloaded periodically by research staff to a central database (Figure 1). Multiple researchers work constantly on data quality and control. Unique "triggers," i.e., anomalies in the time-series data, were used to identify and extract over 8,700 C/NC events from the database. Additionally, 32,500 baseline events were randomly selected for comparison. These events were then reviewed, coded, and evaluated by data reductionists. The coded information enables researchers to easily identify specific events of interest (EOIs).

The initial step to building and testing classification algorithms was to define our event classes. The three event classes of interest are listed below along with their definition using the coded event data in the SHRP2 NDS database.

1) A **drowsy event** is an event from the C/NC or baseline dataset where the driver exhibits obvious signs of being asleep or tired, or is actually asleep while driving, degrading performance of the driving task.
2) A **distracted event** is an event from the C/NC dataset where the driver is not maintaining acceptable attention to the driving task due to engagement in one or more secondary tasks. This is a subjective judgment call by the reductionist indicating whether any secondary tasks the driver might be involved in contributed to the C/NC.
3) An **attentive event** is an event from the baseline dataset where the driver is not engaged in any secondary task. A secondary task is defined as an observable driver engagement not critical to the driving task such as non-driving related glances away from the direction of vehicle movement.

EOIs under each class were pulled from the SHRP2 NDS database for the purposes of this research effort. 571 drowsy events, 1,123 distracted events, and 15,378 attentive events were identified.

B. Data Extraction

1) Quantitative Data: Once the EOIs were identified, time-series data of the corresponding complete trips were retrieved from the SHRP2 NDS database (Figure 1). Epochs were created by extracting data from 65 seconds before the event time to 5 seconds before the event time. We assumed that the driver behavior did not change throughout this 1-minute epoch. The event data consisted of 44 variables and video of the driver's face. Among these 44 variables, 28 of them were raw data directly collected by the DAS in the SHRP2 NDS vehicle, mainly vehicle dynamics (Table I). The rest of the variables were calculated based off the raw variables (Table II).

2) Video data: Video data is captured at a frame rate of 15 fps using a RGB camera facing the driver. Similarly to the other time-series variables, once the EOIs were identified corresponding 30 second video epochs were extracted from the SHRP2 NDS database. The epochs covered from 35 seconds before the event time to 5 seconds before the event time.

C. Data Pre-processing

1) Data Cleaning: All the sensors during data collection were synchronized with respect to the trip clock, which is the first variable listed in Table I. Unfortunately, the timing of the data across variables was asynchronous leading to missing variables at each collection time point. We replaced the missing value for each variable with the last known corresponding value.

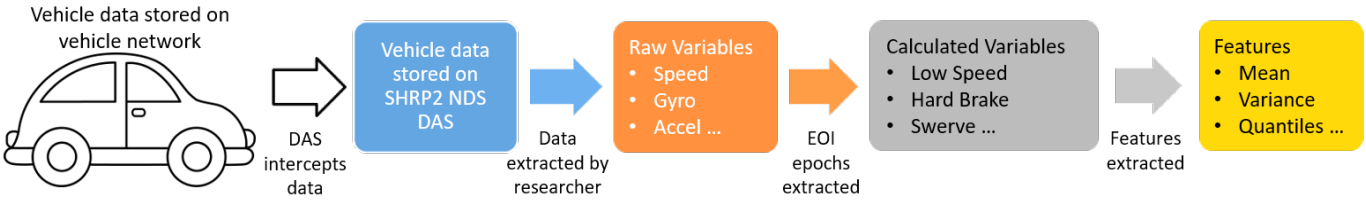

Fig. 1: Data collection, extraction, and pre-processing

Variable Name	Unit	Hz	Note
Timestamp	ms		Time since the beginning of the trip
Speed	km/h	1	Vehicle speed
Gyro z	deg/s	10	Lateral angular velocity
Accel y	g	10	Lateral acceleration
Distance to left lane marker	cm	30	Positive when on the left side of the marker and negative when on the right side
Distance to right lane marker	cm	30	Positive when on the left side of the marker and negative when on the right side
Probability of left marker exist		30	Probability a painted marker exists on the left side of the vehicle's lane
Probability of right marker exist		30	Probability a painted marker exists on the right side of the vehicle's lane
Time of Day	ms	1	UTC time of day
Day		1	From 1 to 31
Month		1	From 1 to 12
Year		1	Last two digits of year
Longitudinal Distance Target 1–8	m	15	Longitudinal distance to radar target 1–8
Lateral Distance Target 1–8	m	15	Lateral distance to radar target 1–8

TABLE I: List of raw variables

Variable Name	Unit	Hz	Note
Timestamp	ms		Time since the beginning of the trip
Variance of Speed		10	Variance of speed of last 30 seconds
Variance of Lane Position		30	Variance of left lane distance of last 30 seconds
Variance of Throttle Position		80	Variance of throttle position of last 30 seconds
Low Speed		1	Indicates the speed is below 30kph: binary
Hard Brake		10	1- 30 seconds after heavy deceleration (0.4g): binary
Day of Week			0 is Sunday, 1 is Monday, etc.
Swerve		10	Indicates if within a 30-second window after a swerve: binary
Passing		15	Passing a vehicle in adjacent lanes: binary
Being Passed		15	Being passed in adjacent lanes: binary
Traffic Flow		15	Indicates if a vehicle is passed more than it is passing: binary
Traffic Level		15	Number of vehicles on radar
Tire Out of Lane		30	Indicates if the vehicle's tire is outside the lane: binary
Lane Change		30	Indicates if within a 30-second window after a lane change: binary
Lane Bust		30	Indicates if within a 30-second window after a lane bust: binary
Time to Line Crossing	sec	30	The time to cross a lane line under current status using lane distance and lateral speed: positive: approaching left line and negative: approaching right line
Active Steering		10	The entire course of steering where peak value exceeds a threshold

TABLE II: List of calculated variables

Feature Name	Feature Description
Mean	
Variance	
Mean Absolute Change	The mean over the absolute differences between subsequent time-series values.
Number of Peaks	Number of peaks seen over last n samples
Percentage of Reoccuring Datapoints	Percentage of unique values, that are present in the time-series more than once.
Sum of Reoccuring Datapoints	Sum of all data points, that are present in the time-series more than once.
Count Above Mean	Number of values in time-series x that are higher than the mean of x
Count Below Mean	Number of values in time-series x that are lower than the mean of x
Longest Strike Above Mean	Length of the longest consecutive subsequence in time-series x that is bigger than the mean of x
Longest Strike Below Mean	Length of the longest consecutive subsequence in time-series x that is smaller than the mean of x
Last Location of Minimum	The relative last location of the minimal value of x
Last Location of Maximum	The relative last location of the maximum value of x.
Quantile	Calculates the q quantile of x.
Binned Entropy	First bin the samples in to k bins. Compute entropy based on percentage of samples in each bin
Spkt Welch Density	Cross power spectral density of the time-series x at different frequencies.
Augmented Dickey Fuller	A hypothesis test which checks whether a unit root is present in a time-series sample.
Kurtosis	The kurtosis of x (calculated with the adjusted Fisher-Pearson standardized moment coefficient G2).
FFT Coefficient	Fourier coefficients of the one-dimensional discrete Fourier Transform for real input by fast fourier transform.
CWT Coeefficient	Continuous wavelet transform for the Ricker wavelet, also known as the Mexican hat wavelet
Time Reversal Assymetry Statistic	Standard time reversal assymetry statistic
Friedrich Coefficients	Coefficients of polynomial h(x), x is the time-series, which has been fitted to the deterministic dynamics of Langevin model
Max Langevin Fixed Points	Largest fixed point of dynamics $\arg\min_x(h(x) = 0)$ estimated from polynomial h(x), which has been fitted to the deterministic dynamics of Langevin model

TABLE III: List of extracted features. We used Tsfresh open source library for extracting these features. [23]

Since the rate of these sensor values was on the order of 10-30hz, the above approximation was considered accurate and reasonable.

2) Feature Extraction: For each epoch we extracted many statistical features coming from the time and frequency domain. A full list of features can be found in Table III. Each epoch was expanded from the 44 collected variables to 4,492 derived variables, or dimensions (Figure 1). Variables with missing or constant variables were then removed resulting in a feature vector of 3,993 variables. Additionally, some of the frequency based statistical features had "NaN's" and were removed, leading to a final 3,500 dimensional feature vector. Finally, the features were z-score normalized.

IV. Research Methodology

Below is a list of the research questions addressed by the team.

1) What data sensors are most helpful in identifying distracted and drowsy drivers? (V-A)
2) How accurate are machine learning methods at identifying distracted and drowsy drivers? (V-B-l-V-D)
3) What machine learning methods are most effective at identifying distracted and drowsy drivers? (V-B–V-D)

A. Principal Component Analysis

In order to identify the variables and sensors that best predict the driver state we performed a Principal Component Analysis (PCA). PCA additionally allowed us to decorrelate the data by removing redundant features given such a long feature vector (1713x3500).

First we determined the optimal number of useful principal components. Prediction results were compared using varying numbers of principal components. Once this was determined we then summed the PCA coefficients of the optimal components for each variable and its associated features. This resulted in a relative rating of a variable's overall ability to predict driver state as compared to other variables.

B. Statistical Approach to Time-Series Classification

As noted at the end of Section III-A the EOIs were unbalanced. One method utilized to account for this was to balance using a random under sampling technique. However, simple under sampling leads to underutilization of the dataset. Therefore, we also tested a method of oversampling a minority class based on Synthetic Minority Oversampling Technique [24]. The minority class is oversampled by creating more samples

Event Class	EOIs Selected			
	NB/SVM Balanced Dataset	NB Unbalanced Dataset	LSTM	CNN
Drowsy	571	571	571	137
Distracted	571	923	571	237
Attentive	602	1998	571	780

TABLE IV: Count of EOIs selected for training and testing classification methods.

using interpolation between the neighbors. Table IV shows the balanced and unbalanced sample sizes utilized by class.

The training and test sets for the balanced data were chosen randomly across classes, with an 80/20% split of samples for training and testing, respectively. The training and test sets for the unbalanced data were chosen uniformly across classes, with a 70/30% split of samples for training and testing, respectively.

Analysis was done first with the unbalanced dataset using Naive Bayes and Support Vector Machine (SVM), one vs all and multiclass. To see the effect of SMOTE, Naive Bayes was then implemented using an unbalanced dataset, with and without SMOTE. Naive Bayes was chosen because of the high dimensionality of the data. Further clarification was added by training a Naive Bayes algorithm with uniform priors (0.33, 0.33, 0.34).

C. Time-Series Classification with LSTM

We implemented LSTM for multivariate classification with attention mechanism. Figure 2 shows our architecture of LSTM for 3-way classification (softmax layer to obtain a probability distribution). Each epoch's time-series variables were first mean normalized by considering the mean of the time-series over the entire training dataset. The time-series data in an entire epoch were down sampled to 1,024 steps using Fourier decimation before feeding to the LSTM network. The 1,024 down-sampled time steps are fed to LSTM with a batch size of 32. Initially a many to one LSTM network architecture was tested but this resulted in suboptimal loss convergence and accuracy. Therefore, we added a fully connected layer to consider intermediate state features at every time step for prediction. A fully connected layer on top of every time step acts as an attention mechanism whereby the network learns to concentrate on discriminative portions of the time segments. The learning rate and number of layers were chosen based on several empirical iterations. Only 571 instances were used from each class to maintain balance. We used 361 samples from each class for training (63/37% training and testing split, respectively). The network was trained with stochastic gradient descent with a learning rate (lr) of 0.004, momentum of 0.9, and an lr decay factor of 0.1 every 10 epochs. The network was trained for 40 epochs (iterations).

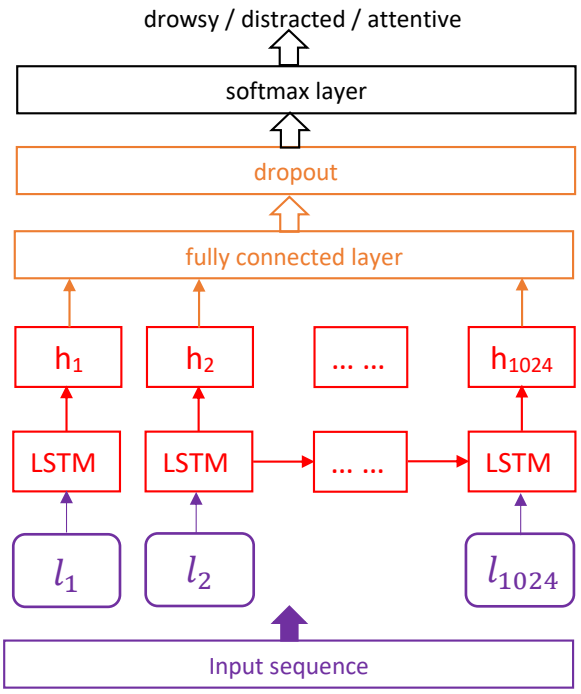

Fig. 2: Block architecture of LSTM. The sequence length is set to 1,024. We use a fully connected layer over all the hidden states to get a weighted feature representation.

D. Classifying Video Epochs Using CNN

We attempted to classify video data epochs using a two stream CNN [25] in PyTorch which employs spatio-temporal feature extraction from static image and optical flow for a segment of frames in a video as represented in Figure 3. The sample dataset consisted of 1,154 total samples. For each of the video epochs, we extracted the optical flow over the 30 second epoch window, indicative of the general movement throughout the video. We chose optical flow as the feature as it captures motion trajectories and a drowsy driver might have slower reaction/movement and head bobbing in contrast to a normal driver. We then randomly sampled 25 frames from the video epochs and computed optical flow between the subsequent frames, which were then fed as input features to the CNN model. One of the issues we found while working with the videos was the limited resolution. The optical flow computation process sometimes introduced grainy artifacts into generated optical flow features. Increasing sampling and decreasing window size might reduce these effects. Additionally, there were issues with optical flow extraction for some drowsy events limiting the availability of an already limited class.

Since our dataset sample size was small and we only extracted a single epoch per trip, we used pre-training followed by a fine tuning procedure for better generalization of CNNs. Initially the two stream CNN was trained on UCF101 and Sports1M datasets and later fine tuned on the selected SHRP2 epochs. Deep Neural Networks perform better when transferred from

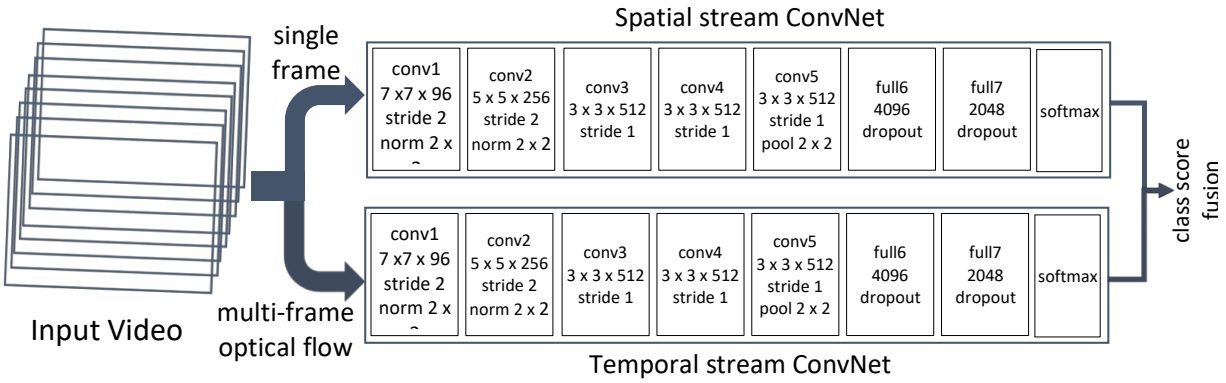

Fig. 3: Model Architecture of CNN video classification

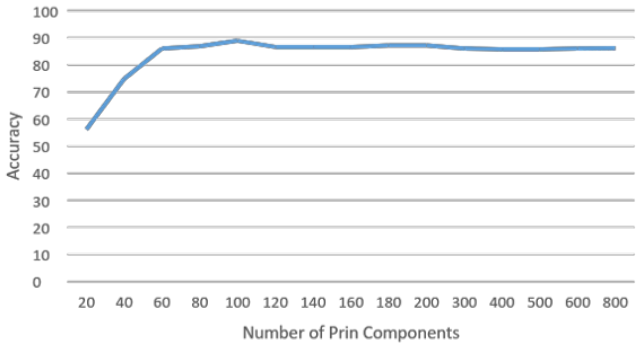

Fig. 4: Comparison of Naive Bayes classification accuracy varying the number of principal components used.

related domains and UCF-101 [25]. The related datasets chosen for training this CNN are action recognition data sets of realistic action video with some facial actions. The fine tuning was completed on a highly skewed set of epochs (Table IV) [25]. The dataset was split for training and testing, 70/30% respectively.

V. Results

This section will discuss the results for each of the methods introduced in the previous section.

A. Principal Component Analysis/Variable Contribution

Figure 4 shows that utilizing more than 100 principal components has an insignificant impact on the accuracy of the prediction algorithm. Therefore, 100 components were used for the remainder of the analysis.

Figure 5 shows the variables that had the most impact, based on their corresponding features, summed across the top 100 components. The variables related to acceleration, speed, and lane marking seem to be the most significant features which makes intuitive sense.

B. Statistical Approach to Time-Series Classification

Table V shows the classification accuracy of Naive Bayes and SVM using a balanced under-sampled dataset.

Classification Method	Accuracy
Naive Bayes	0.87
SVM One vs All	0.90
SVM Multiclass	0.72

TABLE V: Classification accuracy for Naive Bayes and SVM on a balanced dataset.

Figure 6a is the heat map confusion matrix for Naive Bayes without SMOTE using a balanced under-sampled dataset.

Figures 6b through 6d show the confusion matrices for Naive Bayes using the unbalanced dataset. We compare results of this dataset with and without SMOTE as well as using uniform priors.

Based on Figures 6b through 6d it is clear that SMOTE is not just helping the priors but also the modeling of class conditional density.

A summary of the Naive Bayes methods and their results is provided in Table VI. Comparing the results of the balanced dataset (Table VI-a) to the unbalanced dataset (Table VI-d) we see that overall precision, recall, and accuracy improves by leveraging an unbalanced dataset thanks to SMOTE. That being said drowsy event identification was better with the balanced dataset (Figure 6).

1) Validation: In addition to using the originally defined epochs for testing the prediction algorithm we also tested the algorithm against modified epochs to identify the robustness of the algorithm. Modified epochs were created by shifting the time of the data extraction constraints mentioned in Section III-B1. For example, instead of extracting from 65 seconds before to 5 seconds before the event time we extracted from 66 seconds before to 6 seconds before the event time. We assumed that if someone was drowsy during the initially defined time period they were also likely drowsy during the time period shifted back one or two seconds.

One difficulty of creating these new events was how to accomplish normalization. There was a concern that the algorithm was trained on the normalization of the training set and that the parameters would not transfer to the new set of epochs. We

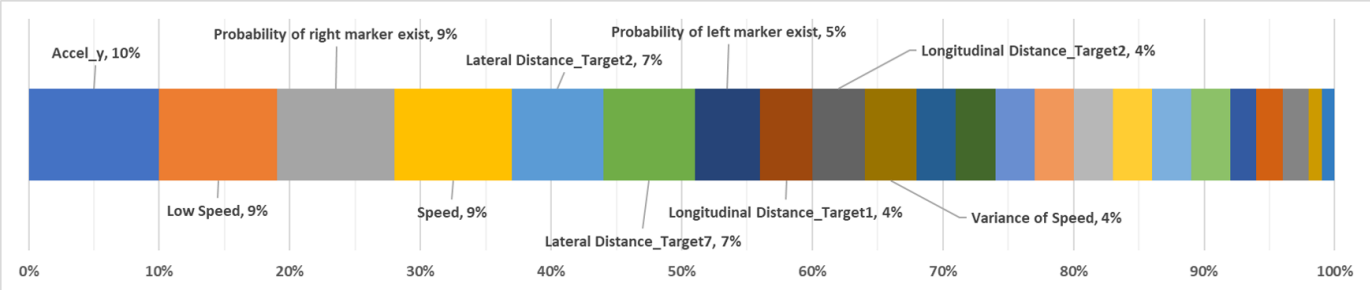

Fig. 5: Top 10 contributors to the top 100 principal components.

Actual Class	Predicted Class		
	Drowsy	Distracted	Attentive
Drowsy	0.95	0.02	0.04
Distracted	0.01	0.80	0.19
Attentive	0.07	0.05	0.88

(a) NB without SMOTE using a balanced dataset
Note: percentages do not sum to one because of rounding

Actual Class	Predicted Class		
	Drowsy	Distracted	Attentive
Drowsy	0.51	0.03	0.46
Distracted	0.01	0.56	0.43
Attentive	0.01	0.05	0.94

(b) NB without SMOTE using an unbalanced dataset

Actual Class	Predicted Class		
	Drowsy	Distracted	Attentive
Drowsy	0.70	0.03	0.27
Distracted	0.01	0.65	0.34
Attentive	0.01	0.05	0.94

(c) NB with Uniform priors using an unbalanced dataset

Actual Class	Predicted Class		
	Drowsy	Distracted	Attentive
Drowsy	0.88	0.04	0.08
Distracted	0.00	0.95	0.05
Attentive	0.00	0.03	0.97

(d) NB with SMOTE using an unbalanced dataset

Fig. 6: Heat map of the Confusion Matrix for each Naive Bayes (NB) method. Scenarios match those in Table VI.

Classification Methodology	**Recall**	**Precision**	**F1 Score**	**Accuracy**
a) Naive Bayes without SMOTE with Balanced Dataset	0.875	0.870	0.873	0.869
b) Naive Bayes without SMOTE with Unbalanced Dataset	0.673	0.821	0.739	0.772
c) Naive Bayes with Uniform Priors with Unbalanced Dataset	0.764	0.855	0.807	0.824
d) Naive Bayes with SMOTE with Unbalanced Dataset	**0.932**	**0.954**	**0.943**	**0.948**

TABLE VI: Summary of results for each method. Recall and precision are calculated as macro-recall and macro-precision.

tested a small number of epochs using Naive Bayes and neither 1) using the original training/test set normalization parameters completed in Sec. III-C2 or 2) creating new normalization parameters for our limited number of epochs was effective in obtaining comparable performance results.

C. Time-Series Classification with LSTM

Initially the network had good learning from the data, however after a few epochs the learning stopped. The best test accuracy with various model configurations and learning rates was 54%. The low accuracy is likely due to the limited dataset size.

D. Classifying Video Epochs Using CNN

The accuracy of the CNN was 70%. Comparatively a weighted baseline prediction model that classifies all epochs as attentive would have approximately a 67% accuracy but recall and precision would be much lower. As mentioned earlier, the limited data sample size was the primary reason for average performance of CNNs. Further research into considering multiple epochs from the same trip could help to alleviate this issue. Ensuring the labels are valid on other epochs would be necessary.

While training the network, we used a fixed learning rate of

10-3 and trained the network for 50 epochs. Further iterations in the hyperparameter selection might get better results.
Finally, we see that even though flow features help in the classification, they are not strongly discriminative among the classes. Flow features that focus on a person's face or track the facial keypoints would provide a better and more discriminative feature.

VI. Conclusion

In this work we focus on several machine learning methods to build a driver monitoring and classification system based on the SHRP2 NDS dataset. Among the methods discussed, the statistical feature extraction method achieved good results with an F1 score of 94% after balancing the data with SMOTE. The LSTM and the video classification CNN had promising results from the limited available dataset, however the performance of these deep learning approaches was not on par with the traditional statistical feature extraction methods. The small dataset size is the primary cause for the average performance of these methods.
From the results it can be inferred that SMOTE helps in better modeling of class conditional densities and improves overall performance.

Future Research

Additional direct features (head pose, gaze, PerClos, activity tracking) as DNN or Time-Series would greatly improve the results. These variables are considered highly indicative of drowsy and distracted drivers but were not currently available in our dataset.
Solving the inconsistencies involved with normalizing additional test sets would additionally, allow for more thorough validation of the seemingly effective statistical feature extraction methods.
To overcome issues with the small training size in LSTM, techniques such as one shot/few shots, meta-learning approaches could be employed. Another solution would be to use semi-supervised learning with Generative Adversarial Networks or autoencoders to learn the entire data distribution of the SHRP2 dataset and then fine tune for classification. Further machine learning methods worth testing are Unsupervised Learning using Clustering, Hidden Markov Models (HMM), and Symbolic Aggregation.
Finally to improve CNN accuracy it would be worth testing samples of more than 25 frames from the video epochs.

Acknowledgment

This work is supported in part by the National Science Foundation via grant IIS-1633363. The US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or the U.S. Government.

References

[1] J. M. Hankey, M. A. Perez, J. A. McClafferty, Description of the shrp 2 naturalistic database and the crash, near-crash, and baseline data sets, Tech. rep., Virginia Tech Transportation Institute.
[2] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, A. Torralba, Eye tracking for everyone, in: IEEE conference on computer vision and pattern recognition, 2016, pp. 2176–2184.
[3] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Appearance-based gaze estimation in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4511–4520.
[4] E. Murphy-Chutorian, M. M. Trivedi, Head pose estimation in computer vision: A survey, IEEE transactions on pattern analysis and machine intelligence 31 (4) (2009) 607–626.
[5] G. Fanelli, J. Gall, L. Van Gool, Real time head pose estimation with random regression forests, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 617–624.
[6] A. Tsuchida, M. S. Bhuiyan, K. Oguri, Estimation of drowsiness level based on eyelid closure and heart rate variability, in: International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2009, pp. 2543–2546.
[7] K. F. Van Orden, T.-P. Jung, S. Makeig, Combined eye activity measures accurately estimate changes in sustained visual task performance, Biological psychology 52 (3) (2000) 221–240.
[8] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, A. Saxena, Car that knows before you do: Anticipating maneuvers via learning temporal driving models, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3182–3190.
[9] O. Olabiyi, E. Martinson, V. Chintalapudi, R. Guo, Driver action prediction using deep (bidirectional) recurrent neural network, arXiv preprint arXiv:1706.02257.
[10] W. Dong, J. Li, R. Yao, C. Li, T. Yuan, L. Wang, Characterizing driving styles with deep learning, arXiv preprint arXiv:1607.03611.
[11] M. Mollenhauer, Z. Doerzaph, M. Song, A. Sarker, Driver fatigue, distraction, and alerting technology phase 2 sbir final report, Tech. rep., USDOT FMCSA Contract DTRT57-10-C-10045.
[12] A. C. Harvey, Forecasting, structural time series models and the Kalman filter, Cambridge university press, 1990.
[13] R. McCleary, R. A. Hay, E. E. Meidinger, D. McDowall, Applied time series analysis for the social sciences, Sage Publications Beverly Hills, CA, 1980.
[14] Y. Matsubara, Y. Sakurai, Regime shifts in streams: Real-time forecasting of co-evolving time sequences, in: KDD, ACM, 2016, pp. 1045–1054.
[15] K. Kalpakis, D. Gada, V. Puttagunta, Distance measures for effective clustering of arima time-series, in: ICDM, IEEE, 2001, pp. 273–280.
[16] S. Papadimitriou, A. Brockwell, C. Faloutsos, Adaptive, hands-off stream mining, in: Proceedings of the 29th International Conference on Very Large Data Bases, 2003, pp. 560–571.
[17] L. Li, B. A. Prakash, C. Faloutsos, Parsimonious linear fingerprinting for time series, Proc. VLDB Endow. 3 (1-2) 385–396.
[18] A. Jain, A. Singh, H. S. Koppula, S. Soh, A. Saxena, Recurrent neural networks for driver activity anticipation via sensory-fusion architecture, in: IEEE International Conference on Robotics and Automation, IEEE, 2016, pp. 3118–3125.
[19] A. Zyner, S. Worrall, E. Nebot, A recurrent neural network solution for predicting driver intention at unsignalized intersections, IEEE Robotics and Automation Letters 3 (3) (2018) 1759–1764.
[20] C. Miyajima, K. Takeda, Driver-behavior modeling using on-road driving data: a new application for behavior signal processing, IEEE Signal Processing Magazine 33 (6) (2016) 14–21.
[21] A. Zyner, S. Worrall, J. Ward, E. Nebot, Long short term memory for driver intent prediction, in: IEEE Intelligent Vehicles Symposium, IEEE, 2017, pp. 1484–1489.
[22] K. L. Campbell, The shrp 2 naturalistic driving study: Addressing driver performance and behavior in traffic safety, TR News 282.
[23] M. Christ, et al., tsfresh, https://tsfresh.readthedocs.io/en/latest/index.html, revision e7f2e568 (2017).
[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.
[25] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in neural information processing systems, 2014, pp. 568–576.