Forward Collision Warning with Machine Learning

Domenick Poster, Student

***Abstract*—In this paper, four common machine learning algorithms (Linear Regression, Support Vector Machine, Decision Tree, and Stochastic Gradient Descent) are trained for the purpose of forward collision warning. Their performance is evaluated against CAMP Linear, a warning algorithm based on static, kinematic calculations of vehicle and driver data. The data used is derived from a large scale naturalistic driving study. The type of data employed by the algorithms to predict collision alerts can be shared over vehicular ad-hoc wireless networks. Of the four machine learning algorithms, Support Vector Machines and Stochastic Gradient Descent are able to match or exceed CAMP Linear's prediction capabilities in terms of both accuracy and precision, especially at high levels of network congestion. The machine learning algorithms also show potential for further improvement given their ability to continually learn and adapt. These promising results indicate further research on machine learning based cooperative collision warning systems should be conducted, as little currently exists.**

Keywords*—machine learning, cooperative collision warning, forward collision warning

I. Introduction

Forward collisions amount to 30% all of crashes nationwide [1]. Recent advances in technology and greater availability of driving data have followed a surge of interest in assisted driving systems aimed at preventing crashes and saving lives. Forward Collision Warning (FCW) algorithms aim to prevent these crashes by issuing a preemptive warning signal. Machine learning has also seen a increase in interest and is being successfully applied to a wide variety of problems. Within the domain of vehicular collision avoidance, computer vision using camera and radar data has become a popular although challenging solution.

Vehicle-to-vehicle communication allows for an alternative. In a Cooperative Collision Warning (CCW) system, cars share pertinent driving data such as position, speed, and acceleration over wireless communication channels. FCW algorithms can analyze this data to calculate whether or not a crash is likely if no preventative actions are taken. However, little research is available studying the performance of machine learning algorithms in a CCW system.

The performance of a FCW algorithm is limited by the accuracy of its input data. In CCW systems, data transmitted over wireless networks is prone to loss and latency. FCW algorithms must be robust to the challenges presented by vehicle-to-vehicle (V2V) communication. When considering assisted driving systems, humans are a major variable. Naturalistic driving studies [2] have allowed for better understanding of human driving behavior and reaction times. This data allows FCW algorithms to incorporate constraints on human ability into the timing of collision alerts.

One approach to designing a FCW algorithm is to essentially treat it as physics problem by merging vehicle data (location, speed, acceleration, braking, etc.) with human reaction times to calculate important kinematic information such as the time to crash or brake onset range. Alternatively, machine learning algorithms can be trained on large datasets of driving trials in order to learn key patterns associated with warnings and collisions. In this manner, machine learning algorithms can decide when to generate alerts without "understanding" the physics of the scenario.

In order to assess the viability of machine learning based FCW systems, this paper evaluates and compares various machine learning algorithms against a gold standard generated by the CAMP Linear algorithm [3]. One linear regression algorithm and three machine learning classification algorithms are evaluated. The three machine learning classification algorithms are Support Vector Machine (SVM), Decision Tree (DT), and Stochastic Gradient Descent (SGD).

II. Relevant Work

The 100-Car Naturalistic Driving Study [2] is a database of recorded driving data containing information such as vehicle speed, acceleration, location, and driver reaction times. The study also covers information collected from crashes and near-crashes. Together this data represents a valuable asset for designing FCW algorithms.

The FCW algorithm developed by the Crash Avoidance Metrics Partnership (CAMP) [3] used data from the 100-car naturalistic driving study [2] and a linear regression model to hone the timing of alerts. As such, it is also referred to as the CAMP Linear algorithm. The algorithm calculates an alert range based upon the given acceleration and velocity of both a following vehicle (FV) and a lead vehicle (LV). When the following vehicle's distance to the lead vehicle closes to within the alert range, a warning is issued.

FCW algorithms can be seen as independent components of a larger CCW system [1]. When modeled in this way, the collision warning system can be decoupled from the specific FCW algorithm. In fact, some systems [4] even constitute a hybrid approach of utilizing both wirelessly transmitted data and onboard computer vision sensors. When abstracted away from the system, FCW algorithms can be comparatively evaluated. In a comparison of three FCW algorithms [5], CAMP Linear [3] achieved the highest overall accuracy. However, all three algorithms suffered from being overly cautious and generating too many alerts. Other studies have looked at alternative algorithms based on kinematic calculations [6][7], but little research is available evaluating machine learning based FCW algorithms.

III. Performance Evaluation

A. Vehicle Driving Data

Data of the lead vehicle (LV) is taken from the 100-Car naturalistic driving study [2]. Imperfect network data for the LV is obtained by simulating data received over a wireless network with a given sampling rate and packet error rate (PER). Following vehicle (FV) data is constructed using the Microscopic Traffic Simulator (MITSIM) car-following model.

The algorithms evaluated in this paper operate on the acceleration and velocity of the LV and FV, as well vehicle separation distance. Vehicle separation distance is derived from the positional data of the LV and FV. Vehicle data is sampled every .1 seconds and used to determine if a warning should be issued. Times when the vehicles are further than 200 meters apart are ignored. Data from 823 driving trials with a total of 364,531 time samples is used for training and testing.

B. Machine Learning Algorithms

The implementations of the machine learning algorithms are provided by the Scikit-learn Python machine learning library [8]. Figure 1 demonstrates the primary differences between the machine learning linear regression approach to FCW versus the machine learning classification approach. The linear regression (LR) algorithm takes as input the acceleration and velocity of the LV and FV, the estimated vehicle separation distance, and training values corresponding to the expected warning range. The LR algorithm then predicts a warning range based on unseen acceleration, velocity, and separation distance data points.

Conversely, Support Vector Machine (SVM), Decision Tree (DT), and Stochastic Gradient Descent (SGD) are machine learning classification algorithms. Their binary output represents whether or not a warning signal should be issued. Again, the algorithms are trained on acceleration, velocity, and separation distance. However instead of being trained on expected warning ranges, they are trained on the binary warning labels from the ground truth data.

C. Methodology

Unfortunately, this dataset lacks a ground truth for when a warning needs to be generated. Instead, warnings generated by the CAMP algorithm under perfect network conditions (0% PER) must suffice as the ground truth. Therefore, the performance of the algorithms are always judged against the performance of CAMP Linear under perfect network conditions.

While many FCW performance evaluation metrics exist [9], such as the percentage of the population who would be expected to avoid the crash given the timing of an alert, the metrics examined throughout this paper are derived from ratios of true positives, false negatives, etc. For the purpose of this paper, a positive response is when a warning is generated. A negative response indicates a safe scenario (no warning). The accuracy of a system is derived from the ratio of true positives and true negatives to all outcomes. In other words, the accuracy is how often the system output matches the ground truth.

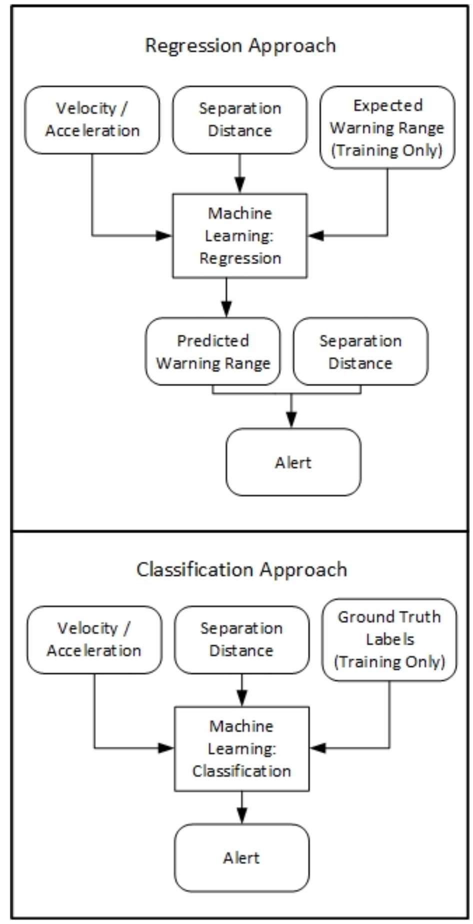

Fig. 1. Regression vs Classification

Precision corresponds to the ratio of true positives over both true positives and false positives. A high precision means more correct warnings and fewer false alarms. The false negative rate (FNR) describes the percentage of missed warnings. A high FNR means a greater likelihood of a driver not receiving a warning in time to avoid an impending crash. The false positive rate (FPR) gives an indication of how sensitive or cautious the algorithm is. A high FPR means the driver receives many unnecessary warnings. Figure 2 provides a visual illustration of these relationships.

The algorithms are evaluated under multiple scenarios. The first set of experiments test the performance of the machine learning algorithms when trained on perfect network data (0% PER). 40% of the data is used for training and 60% is reserved for testing. Trials are run on testing data received from a simulated wireless network with 0%, 10%, 30%, 60%, and 90% PER.

Next, the algorithms are trained on imperfect network data (10%, 30%, 60%, and 90% PER) and tested on data received under the same PER conditions. Again, 40% of the data is used for training and 60% for testing.

Finally, the algorithms' performance are examined with varying sizes of training data. 1%, 5%, 10%, 20%, 30%, 40%,

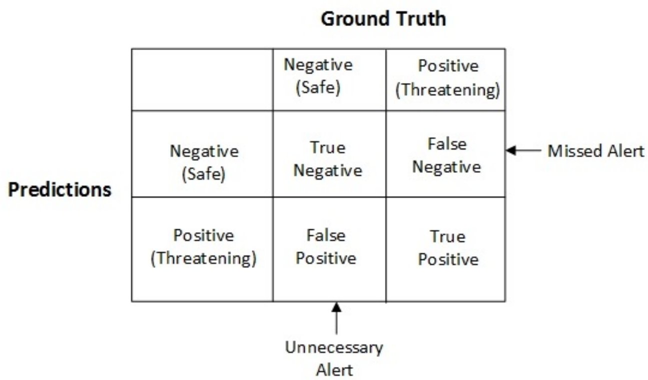

Fig. 2. FCW Confusion Matrix

and 60% of the data is used for training in the different trials. 0% PER network data is used for the training and 30% PER network data is used for testing throughout this last set of experiments.

All results are validated with a 5-fold cross-validation scheme where training and testing data is chosen by random sampling. The graphs are created with the use of the matplotlib [10] Python library.

IV. Results and Discussion

A. Accuracy

In the first series of experiments, the machine learning algorithms were trained on perfect network data. PER values of 0%, 10%, 30%, 60%, and 90% are used for the testing data. Figure 3 illustrates the results. In terms of accuracy,

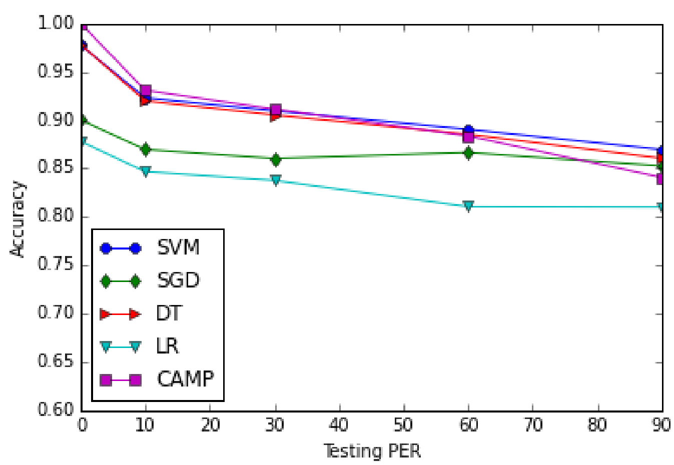

Fig. 3. Accuracy with 0% PER Train Data, Variable PER Test Data

Linear Regression (LR) is clearly the worst performer of the four, with Stochastic Gradient Descent (SGD) coming in 3rd. The Support Vector Machine (SVM) and Decision Tree (DT) algorithms closely track the performance of the CAMP Linear algorithm. However, as the network PER increases beyond 30%, SVM and DT actually slightly outperform CAMP Linear.

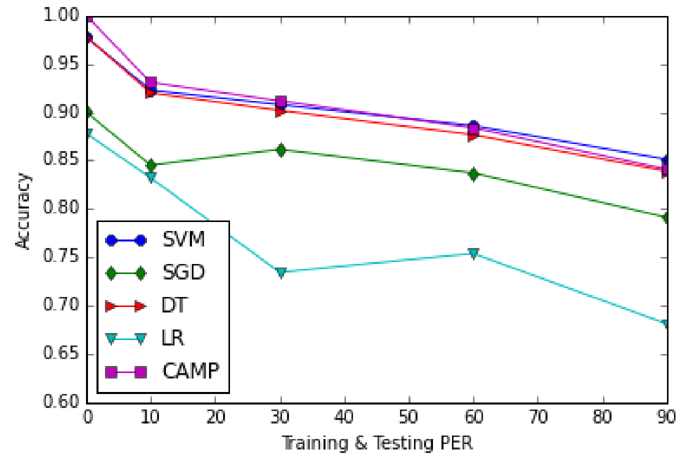

Fig. 4. Accuracy with Equal Train and Test PER

Figure 4 plots accuracy when the machine learning algorithms are trained and tested on data received with the same PER. In other words, if training is done with 30% PER, then testing is also conducted with 30% PER. In this scenario, the machine learning algorithms fail to gain an edge on CAMP Linear at higher PER values. SVM and DT remain the top performers.

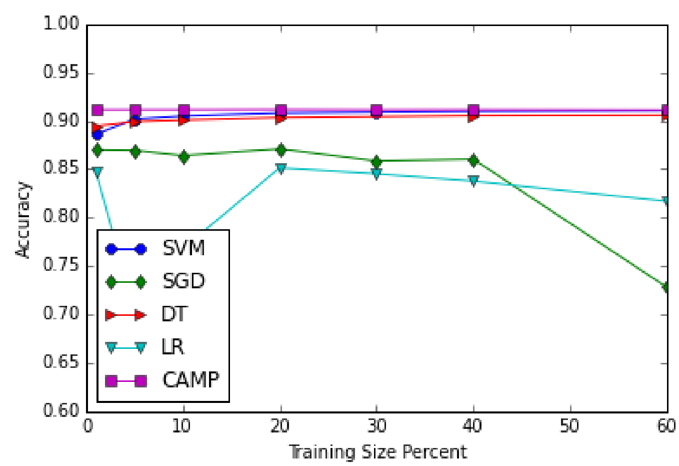

Fig. 5. Accuracy with Variable Training Size Percentages

Figure 5 shows the lack of impact training size has on accuracy. Training was conducted on perfect network data while testing data was generated with a 30% PER. Even when using only 1% of the data, the algorithms still have access to over 3,600 training samples. With only 10% of data for training, SVM and DT achieve performance comparable with CAMP Linear. Additional training data does little to further boost performance. Essentially, the algorithms are bound by the ground truths generated by CAMP Linear, and thus struggle to exceed CAMP performance. More sophisticated training data may help to improve accuracy beyond CAMP Linear's capabilities.

B. Precision

When looking at precision (see Figure 6), which is the ratio of correct warnings to all predicted warnings, SVM consistently scores higher than CAMP Linear. These results are from training done on perfect network data. Similar to the accuracy scores, the machine learners do not exceed CAMP Linear performance when training on imperfect network data (Figure 7). DT only begins to outperform CAMP Linear at higher PER values. Corroborating results from [9], CAMP Linear appears to be overly sensitive.

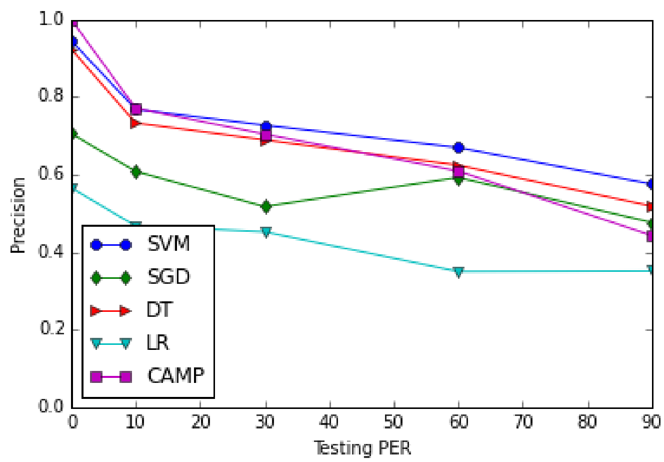

Fig. 6. Precision with 0% PER Train Data, Variable PER Test Data

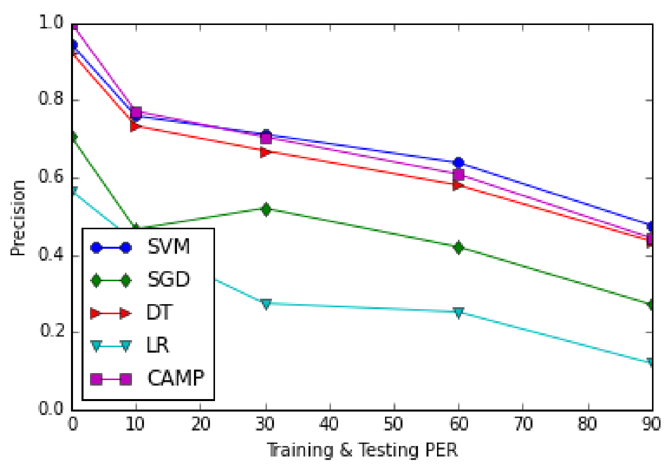

Fig. 7. Precision with Equal Train and Test PER

C. False Positives and Negatives

Even though accuracy and precision capture a sense of the reliability and sensitivity of the algorithm, the false negative rate (FNR) and false positive rate (FPR) are presented here to reinforce the findings. From Figures 8 and 9, one can see the trade-off between receiving more legitimate warnings versus receiving fewer false alarms. While CAMP Linear is more sensitive (a higher FPR), it is rewarded with fewer missed alerts (lower FNR). Granted, machine learning algorithms can be tweaked to more heavily weight warnings during the training process. Given the nature of the FCW scenario, this may be desirable.

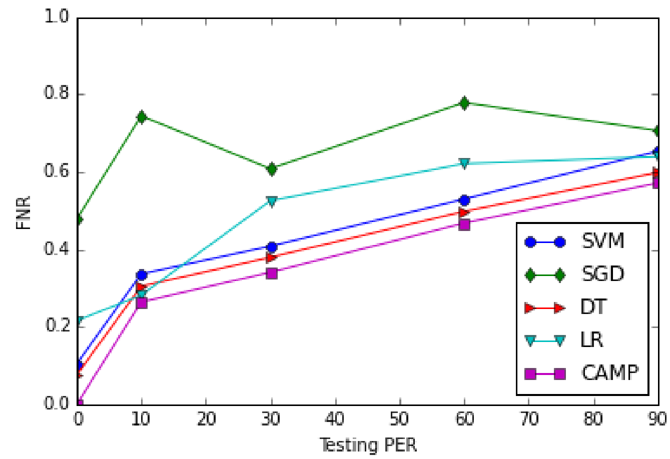

Fig. 8. FNR with 0% PER Train Data, Variable PER Test Data

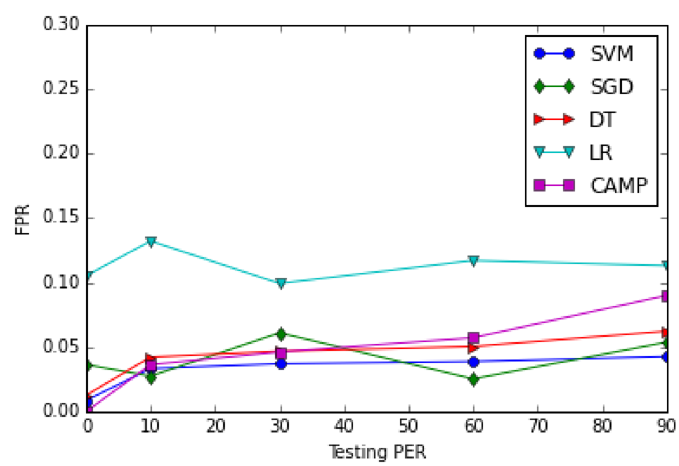

Fig. 9. FPR with 0% PER Train Data, Variable PER Test Data

V. Conclusion

This paper demonstrates the viability of certain machine learning algorithms to address the FCW problem. SVM and

DT achieve similar performance to CAMP Linear when network PER is low. If SVM and DT are trained on network data with 0% PER, they outperform CAMP Linear in both accuracy and precision as wireless network congestion increases. When training on imperfect network data, the machine learning algorithms at best track CAMP Linear's performance.

CAMP Linear is shown here to be more cautious and sensitive compared to the machine learning algorithms. However, machine learning classifiers can be trained to be more biased toward erring on the side caution. Given the massive datasets generated by naturalistic driving studies and the ability of machine learning algorithms to continually adapt, there should be no shortage of training data to leverage for better performance or customization of warnings to individual drivers. While prediction times observed during this study were very fast (on the order of microseconds), training was a time consuming process, especially for SVM. This would need to be taken into account when deployed in a vehicle.

With more sophisticated training and tweaking of machine learning algorithms, their performance may be able to exceed traditional kinematic predictions. Further work should be directed towards creating more advanced training sets. For example, a more reliable or objective ground truth for training besides the gold standard generated by CAMP Linear under perfect network conditions should be established. Also, training should take into account the time-series nature of the data instead of treating the data as isolated slices of time as done in this paper. These measures would help allow for more domain-specific performance evaluation metrics beyond accuracy and precision.

References

[1] Y. P. Fallah and M. K. Khandani, "Analysis of the coupling of communication network and safety application in cooperative collision warning systems," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, pp. 228–237, ACM, 2015.

[2] T. A. Dingus, S. Klauer, V. Neale, A. Petersen, S. Lee, J. Sudweeks, M. Perez, J. Hankey, D. Ramsey, S. Gupta, *et al.*, "The 100-car naturalistic driving study, phase ii-results of the 100-car field experiment," tech. rep., 2006.

[3] R. Kiefer, M. Cassar, C. Flannagan, D. LeBlanc, M. Palmer, R. Deering, and M. Shulman, "Forward collision warning requirements project: Refining the camp crash alert timing approach by examining" last-second" braking and lane change maneuvers under various kinematic conditions," tech. rep., 2003.

[4] B. R. Chang, H. F. Tsai, and C.-P. Young, "Intelligent data fusion system for predicting vehicle collision warning using vision/gps sensing," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2439–2450, 2010.

[5] S. B. McLaughlin, J. M. Hankey, T. A. Dingus, and S. G. Klauer, "Development of an fcw algorithm evaluation methodology with evaluation of three alert algorithms," tech. rep., 2009.

[6] V. Milanés, J. Pérez, J. Godoy, and E. Onieva, "A fuzzy aid rear-end collision warning/avoidance system," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9097–9107, 2012.

[7] K. Lee and H. Peng, "Evaluation of automotive forward collision warning and collision avoidance algorithms," *Vehicle System Dynamics*, vol. 43, no. 10, pp. 735–751, 2005.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] S. B. McLaughlin, J. M. Hankey, and T. A. Dingus, "A method for evaluating collision avoidance systems using naturalistic driving data," *Accident Analysis & Prevention*, vol. 40, no. 1, pp. 8–16, 2008.

[10] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

Appendix
Data

TABLE I. Accuracy with 0% PER Train Data, Variable PER Test Data

	0%	10%	30%	60%	90%
CAMP	1.0	0.931	0.912	0.884	0.841
DT	0.978	0.92	0.905	0.885	0.861
SVM	0.978	0.923	0.91	0.891	0.87
LR	0.878	0.847	0.838	0.811	0.81
SGD	0.9	0.87	0.861	0.867	0.853

TABLE II. Precision with 0% PER Train Data, Variable PER Test Data

	0%	10%	30%	60%	90%
CAMP	1.0	0.772	0.704	0.61	0.443
DT	0.924	0.733	0.69	0.624	0.519
SVM	0.946	0.769	0.728	0.67	0.576
LR	0.566	0.468	0.453	0.351	0.352
SGD	0.706	0.609	0.518	0.593	0.477

TABLE III. FNR with 0% PER Train Data, Variable PER Test Data

	0%	10%	30%	60%	90%
CAMP	0.0	0.265	0.341	0.468	0.572
DT	0.076	0.305	0.38	0.498	0.599
SVM	0.105	0.337	0.408	0.53	0.654
LR	0.216	0.282	0.526	0.622	0.639
SGD	0.478	0.745	0.608	0.779	0.707

TABLE IV. FPR with 0% PER Train Data, Variable PER Test Data

	0%	10%	30%	60%	90%
CAMP	0.0	0.036	0.046	0.057	0.09
DT	0.013	0.042	0.047	0.05	0.062
SVM	0.009	0.033	0.037	0.039	0.043
LR	0.105	0.132	0.099	0.117	0.113
SGD	0.036	0.027	0.061	0.025	0.054

TABLE V. TPR WITH 0% PER TRAIN DATA, VARIABLE PER TEST DATA

	0%	10%	30%	60%	90%
CAMP	1.0	0.735	0.659	0.532	0.428
DT	0.924	0.695	0.62	0.502	0.401
SVM	0.895	0.663	0.592	0.47	0.346
LR	0.784	0.718	0.474	0.378	0.361
SGD	0.522	0.255	0.392	0.221	0.293

TABLE VI. TNR WITH 0% PER TRAIN DATA, VARIABLE PER TEST DATA

	0%	10%	30%	60%	90%
CAMP	1.0	0.964	0.954	0.943	0.91
DT	0.987	0.958	0.953	0.95	0.938
SVM	0.991	0.967	0.963	0.961	0.957
LR	0.895	0.868	0.901	0.883	0.887
SGD	0.964	0.973	0.939	0.975	0.946

TABLE VII. ACCURACY WITH EQUAL TRAIN AND TEST PER

	0%	10%	30%	60%	90%
CAMP	1.0	0.931	0.912	0.884	0.841
DT	0.978	0.92	0.902	0.877	0.839
SVM	0.978	0.923	0.908	0.886	0.851
LR	0.878	0.832	0.735	0.754	0.681
SGD	0.9	0.845	0.862	0.837	0.791

TABLE VIII. PRECISION WITH EQUAL TRAIN AND TEST PER

	0%	10%	30%	60%	90%
CAMP	1.0	0.771	0.705	0.609	0.444
DT	0.924	0.734	0.67	0.581	0.436
SVM	0.946	0.76	0.712	0.639	0.476
LR	0.566	0.434	0.275	0.253	0.12
SGD	0.706	0.467	0.521	0.421	0.272

TABLE IX. FNR WITH EQUAL TRAIN AND TEST PER

	0%	10%	30%	60%	90%
CAMP	0.0	0.265	0.34	0.468	0.57
DT	0.076	0.303	0.375	0.494	0.579
SVM	0.105	0.324	0.401	0.525	0.624
LR	0.216	0.477	0.501	0.626	0.811
SGD	0.478	0.447	0.585	0.638	0.731

TABLE X. FPR WITH EQUAL TRAIN AND TEST PER

	0%	10%	30%	60%	90%
CAMP	0.0	0.036	0.046	0.057	0.09
DT	0.013	0.042	0.052	0.061	0.091
SVM	0.009	0.036	0.04	0.045	0.069
LR	0.105	0.115	0.225	0.183	0.235
SGD	0.036	0.106	0.064	0.083	0.121

TABLE XI. TPR WITH EQUAL TRAIN AND TEST PER

	0%	10%	30%	60%	90%
CAMP	1.0	0.735	0.66	0.532	0.43
DT	0.924	0.697	0.625	0.506	0.421
SVM	0.895	0.676	0.599	0.475	0.376
LR	0.784	0.523	0.499	0.374	0.189
SGD	0.522	0.553	0.415	0.362	0.269

TABLE XII. TNR WITH EQUAL TRAIN AND TEST PER

	0%	10%	30%	60%	90%
CAMP	1.0	0.964	0.954	0.943	0.91
DT	0.987	0.958	0.948	0.939	0.909
SVM	0.991	0.964	0.96	0.955	0.931
LR	0.895	0.885	0.775	0.817	0.765
SGD	0.964	0.894	0.936	0.917	0.879