Fraud Detection Using Reputation Features, SVMs, and Random Forests

Dave DeBarr, and Harry Wechsler, *Fellow, IEEE*
Computer Science Department
George Mason University
Fairfax, Virginia, 22030, United States
{ddebarr, wechsler}@gmu.edu

***Abstract*—Fraud is the use of deception to gain some benefit, often financial gain. Examples of fraud include insurance fraud, credit card fraud, telecommunications fraud, securities fraud, and accounting fraud. Costs for the affected companies are high, and these costs are passed on to their customers. Detection of fraudulent activity is thus critical to control these costs. Last but not least, in order to avoid detection, fraudsters often change their "signatures" (methods of operation). We propose here to address insurance fraud detection via the use of reputation features that characterize insurance claims and ensemble learning to compensate for varying data distributions. We replace each of the original features in the data set with 5 reputation features (RepF): 1) a count of the number of fraudulent claims with the same feature value in the previous 12 months, 2) a count of the number of months in the previous 12 months with a fraudulent claim with the same feature value, 3) a count of the number of legitimate claims with the same feature value in the previous 12 months, 4) a count of the number of months in the previous 12 months with a legitimate claim with the same feature value, and 5) the proportion of claims with the same feature value which are fraudulent in the previous 12 months. Furthermore we use two one-class Support Vector Machines (SVMs) to measure the similarity of the derived reputation feature vector to recently observed fraudulent claims and recently observed legitimate claims. The combined reputation and similarity features are then used to train a Random Forest classifier for new insurance claims. A publicly available auto insurance fraud data set is used to evaluate our approach. Cost savings, the difference in cost for predicting all new insurance claims as non-fraudulent and predicting fraud based on a trained data mining model, are used as our primary evaluation metric. Our approach shows a 13.6% increase in cost savings compared to previously published state of the art results for the auto insurance fraud data set.**

Keywords—Fraud Detection; Reputation Features; One Class Support Vector Machine; Random Forest; Cost Sensitive Learning

I. Introduction

According to the most recent Federal Bureau of Investigation Financial Crimes Report to the Public [15], there is an upward trend among many forms of financial crimes including health care fraud. Estimates of fraudulent billings to health care programs, both public and private, are estimated to be between 3 and 10 percent of total health care expenditures. This estimate is consistent with the most recent Association of Certified Fraud Examiners Report to the Nations [2], which showed survey participants estimated the typical organization loses 5% of its revenue each year.

Common types of fraud include tax fraud [11], securities fraud [5], health insurance fraud [18], auto insurance fraud [19, 21], credit card fraud [9], and telecommunications fraud [4, 14]. For tax fraud, a taxpayer intentionally avoids reporting income or overstates deductions. For securities fraud, a company may misstate values on financial reports. For insurance fraud, the insured files claims that overstate losses. For credit card fraud, the credit card is used by someone other than the legitimate owner. Challenges for fraud detection include imbalanced class distributions, large data sets, class overlap, and the lack of publicly available data.

The novelty of our approach to fraud detection is the use of reputation features and one-class SVM similarity features for fraud detection. Reputation features have been used in [1] to analyze the previous behavior of Wikipedia editors for vandalism detection. For fraud detection, reputation features are used to characterize how often feature values from a claim have been associated with fraud in the past. Similarity features, derived from one-class SVMs, are then used to extend reputation from individual features to the joint distribution of features for a claim. Finally, a cost-sensitive Random Forest classification model is constructed to classify new claims based on reputation and similarity features.

Unfortunately there is not much publicly available fraud detection data available for research. Corporate victims of fraud are often reluctant to admit that they have been the victims of fraud, and transactional data is often sensitive (e.g. containing personally identifiable account information). The auto insurance fraud data set used in this study is the only publicly available fraud detection data set that we are aware of.

The remainder of this paper is organized as follows. Section II describes cost sensitive learning. Section III describes reputation and similarity features. Section IV describes the Random Forest classification algorithm. Section V describes our experimental design using the publicly available auto insurance fraud data set. Section VI provides experimental results, and Section VII provides conclusions.

II. Cost Sensitive Learning

The presence of an imbalanced class distribution is a common characteristic for fraud detection applications [5, 17], because fraudulent transactions occur much less frequently than non-fraudulent transactions. For some domains, fraud may occur 10 or more times less frequently than non-fraudulent transactions. Because there are relatively few fraudulent transactions compared to non-fraudulent transactions, larger data sets are required to learn to confidently distinguish fraudulent transactions from non-fraudulent transactions.

To make matters worse, fraudulent transactions often look like non-fraudulent transactions because the fraudsters want to avoid detection; i.e. the fraudulent and non-fraudulent classes overlap. Because fraudulent transactions look like non-fraudulent transactions (the classes overlap), standard pattern recognition "learning" algorithms will make fewer errors by simply declaring all transactions to be non-fraudulent. The resulting classification model is known as a "majority" classifier, because it simply declares all transactions to belong to the majority class (non-fraudulent transactions). Additionally, fraudsters may adapt their observed behavior in response to detection [5]. This leads to an adversarial game in which detection advocates must somehow adapt to changes made by fraudsters, which in turn leads fraudsters to adapt to changes made by detection advocates.

Consider two overlapping distributions: a bivariate Gaussian distribution (with 2 independent features) centered at (2,2) with a standard deviation of 0.5, and a second bivariate Gaussian distribution (with 2 independent features) centered at (0,0) with a standard deviation of 2. Suppose that the prior probability for the class centered at (2,2) is 1% and the prior probability for the class centered at (0,0) is 99%. In this situation, a majority classifier would be the optimal Bayes classifier [13] if the misclassification costs are equal! Bayesian risk for predicting class α_i for observation $\mathbf{x}$ is computed as:

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{C} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid \mathbf{x}) \quad (1)$$

where $\lambda(\alpha_i \mid \omega_j)$ is the loss (misclassification cost) associated with predicting class α_i when the actual class is ω_j and

$$P(\omega_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_j) P(\omega_j)}{\sum_{k=1}^{C} p(\mathbf{x} \mid \omega_k) P(\omega_k)} \quad (2)$$

where $P(\omega_j \mid \mathbf{x})$ is the posterior probability of observation $\mathbf{x}$ belonging to class ω_j, $p(\mathbf{x} \mid \omega_j)$ is the likelihood (density estimate) of observing $\mathbf{x}$ within class ω_j, $P(\omega_j)$ is the prior probability of observing class ω_j, C is the number of classes (2 for fraud detection), and $\sum_{k=1}^{C} p(\mathbf{x} \mid \omega_k) P(\omega_k)$ is the evidence for observation $\mathbf{x}$.

Further suppose that the minority class represents fraud. The principal costs for fraud are the cost of investigations and the cost of paying claims. If an investigation costs \$100 and a claim costs \$5000, then the decision boundary for the optimal Bayes classifier is shown by the solid line in Figure 1. 95% of the fraudulent class distribution lies in the small dotted circle, while 95% of the non-fraudulent class distribution lies in the large dotted circle.

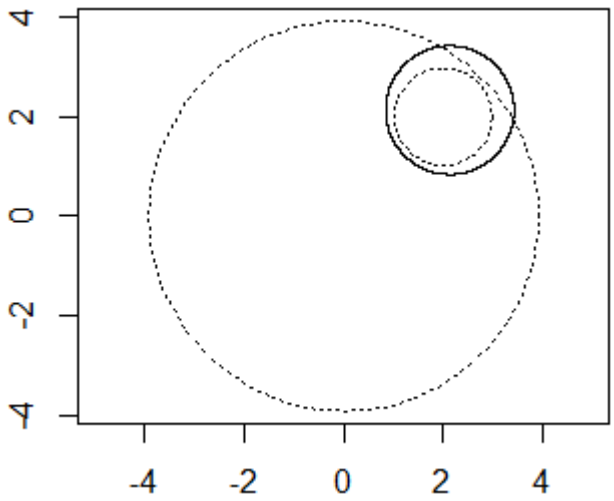

Figure 1. Optimal Bayes Decision Boundary

As shown in Table 1, this classifier would misclassify 4.4% of the fraudulent class distribution as non-fraudulent and 6.8% of the non-fraudulent class distribution as fraudulent; but overall costs would be minimized.

		Predict	
		Fraud	Not Fraud
Actual	Fraud	95.6%	4.4%
	Not Fraud	6.8%	93.2%

Table 1. Optimal Bayes Classification Errors

Strategies for overcoming the tendency to produce a simple "majority" classifier include sampling or cost sensitive learning [12, 17]. For sampling strategies, the training examples are "stratified" (partitioned) into two groups: fraudulent training examples and non-fraudulent training examples. Sampling from the training set can be performed "with" or "without" replacement. In sampling "with" replacement, each training example can be selected more than once, while in sampling "without" replacement, each training example can be selected at most once. In order to balance the fraudulent and non-fraudulent training examples, either the majority class can be under-sampled or the minority class can be over-sampled. Under-sampling

involves selecting a subset of the non-fraudulent training examples, while over-sampling involves selecting a superset of the fraudulent training examples. Selecting a superset of the fraudulent training examples can be performed by sampling with replacement, or even synthesizing new fraudulent training examples similar to known fraudulent training examples [8].

In cost sensitive learning, the training examples are weighted to reflect different misclassification costs. Fraudulent training examples are given a larger weight than the non-fraudulent training examples, reflecting the notion that misclassifying a fraudulent example as non-fraudulent has a higher cost than misclassifying a non-fraudulent training example as fraudulent. This can be viewed as an alternative form of a sampling strategy, where the use of larger weights is a form of over-sampling and the use of smaller weights is a form of under-sampling. Strategies for handling imbalanced class problems can be used with any pattern recognition algorithm, including decision trees, rules, neural networks, Support Vector Machines, and others. This includes the use of bagging and boosting with the MetaCost framework [12].

III. Reputation And Similarity Features

The training process for the proposed approach to fraud detection is illustrated in Figure 2. As shown, our first step is to compute reputation features.

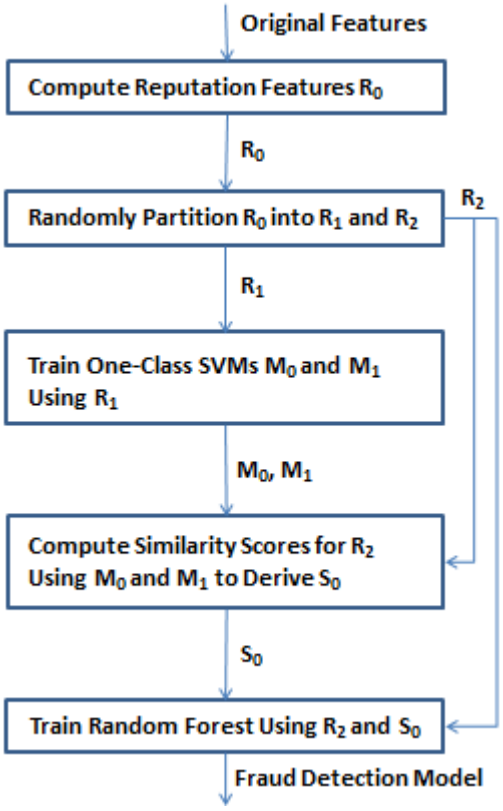

Figure 2. Training Process

We propose replacing each of the original features in the data set with 5 reputation features:

1. Fraud Count: a count of the number of fraudulent claims with the same feature value in the previous 12 months,
2. Fraud Months: a count of the number of months in the previous 12 months with a fraudulent claim with the same feature value,
3. Legitimate Count: a count of the number of legitimate claims with the same feature value in the previous 12 months,
4. Legitimate Months: a count of the number of months in the previous 12 months with a legitimate claim with the same feature value, and
5. Fraud Rate: the proportion of claims with the same feature value which are fraudulent in the previous 12 months.

These 5 values capture support and confidence values for each feature: how often a particular value is observed for a class (support) and what proportion of the time a particular value is associated with fraud (confidence). For the proportion feature we use a Wilson estimate [22] of the proportion to avoid the extremes of zero or one when we have not observed the value very often in previous months. The Wilson estimate is a weighted average of the observed proportion and one half:

$$\hat{p} = \frac{1*\left(\frac{FraudCount}{FraudCount + LegitimateCount}\right) + \frac{1.96^2}{TotalCount} * \left(\frac{1}{2}\right)}{1 + \frac{1.96^2}{TotalCount}} \quad (3)$$

A training set is then randomly partitioned into two equal size subsets. The first subset is used to derive two one-class SVMs, while the second subset is used to construct a Random Forest classifier using the reputation and one-class SVM similarity features.

The two one-class Support Vector Machines (SVMs) measure the similarity of the derived feature vector to previously observed fraudulent claims and previously observed legitimate claims. One class SVMs [20] are used to estimate the probability of class membership. Given a set of observations $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$, a one class SVM is trained by finding the corresponding α_i coefficient for each training observation such that the following expression is minimized:

$$\min_{\alpha}\left(\frac{1}{2}\alpha^T \mathbf{Q}\alpha\right) \quad (4)$$

subject to the following constraints:

$$0 \le \alpha_i \le 1 \quad (5)$$

$$\sum_{i=1}^{n} \alpha_i = \nu n \quad (6)$$

where

$$\mathbf{Q}_{i,j} = K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \phi\left(x_i\right)^T \phi\left(x_j\right) \quad (7)$$

Training observations with a non-zero α_i coefficient are known as "support vectors", because they define the decision

boundary. The kernel function K is used to measure similarity of two observations, which is the equivalent of measuring the dot product in a higher dimension feature space. The radial basis function was used as the kernel function:

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{\sigma^2}\right) \quad (8)$$

where σ is the mean of the 10th, 50th, and 90th percentile of Euclidean distance values for a random sample of *n*/*2* pairs of observations [7]. The hyper-parameter ν places an upper bound on the proportion of the training data that can be declared to be outliers and a lower bound on the proportion of the training set to be used as support vectors. The value of ν was chosen to be 0.05.

Once the α_i coefficients have been found, the distance of a new observation from the class boundary defined by the one-class SVM can be computed as:

$$\sum_{i=1}^{n}\left(\alpha_i K\left(\mathbf{x}, \mathbf{x}_i\right)\right) - \rho \quad (9)$$

where ρ is chosen as the offset that yields $1 - \nu n$ positive values for observations of the training set. The similarity feature of the one-class SVM is a measure of how well a new observation fits with the observed training distribution. The larger the similarity feature value, the more likely the observation belongs to the distribution characterized by the training data. The probability of membership can be estimated by comparing the distance for a new observation to the distance values computed for the training data.

To generate 2 new features for input to a classification model, two one-class SVMs are constructed using the first subset of training data. The first one-class SVM is constructed from fraudulent transactions in the first subset of training data, while the second one-class SVM is constructed from non-fraudulent transactions in the first subset of training data. Unlike the other reputation features, the one-class SVM similarity features consider the joint distribution of feature values when evaluating feature vectors.

IV. Random Forests

The derived feature vectors for the second subset of training data, including the two one-class SVM similarity features, are used to construct a Random Forest classifier [6]. The Random Forest algorithm is an implementation of bootstrap aggregation (bagging) where each tree in an ensemble of decision trees is constructed from a bootstrap sample of feature vectors from the training data. Each bootstrap sample of feature vectors is obtained by repeated random sampling with replacement until the size of the bootstrap sample matches the size of the original training subset. This helps to reduce the variance of the classifier (reducing the classifier's ability to overfit the training data). When constructing each decision tree, only a randomly selected subset of features is considered for constructing each decision node. Of the *k* randomly selected features to consider for constructing each decision node, the yes/no condition that best reduces the Gini impurity measure *g* of the data is selected for the next node in the tree:

$$g = 1 - P\left(Fraud\right)^2 - P\left(NotFraud\right)^2 \quad (10)$$

The Gini impurity measure is largest when the classifier is most uncertain about whether a feature vector belongs to the fraud class.

To support cost sensitive learning, we used a balanced stratified sampling approach [10] to generate bootstrap samples for training the classifier. For training each tree, a bootstrap sample is drawn from the minority class and a sample of the same size is drawn (with replacement) from the majority class. This effectively under-samples the majority class.

To classify new feature vectors, the reputation features and two one-class SVM similarity features are derived, then each tree in the Random Forest classification model casts its vote for a class label: fraud or not fraud. The proportion of votes for the fraud class is the probability that a randomly selected tree would classify the feature vector as belonging to the fraud class. This is interpreted as the probability of a feature vector belonging to the fraud class.

V. Experimental Design

The auto insurance data set [3] has been used to demonstrate fraud detection capabilities [16, 19]. As this is the only publicly available fraud data set, we use it for our experiments as well. It consists of 3 years of auto insurance claims: 1994, 1995, and 1996. Table 2 describes the distribution of fraud and not-fraud claims by year.

Year	Fraud	Not Fraud	Fraud Rate
1994	409	5,733	6.7%
1995	301	4,894	5.8%
1996	213	3,870	5.2%
All	923	14,497	6.0%

Table 2. Fraud Rates for Auto Insurance Data

The proportion of overall claims that are fraudulent is only 6%, so only 1 in 17 claims are fraudulent. Table 3 lists the features of the data. As shown, two of the features were not used for prediction. The Year attribute obviously does not generalize to future data. As we assume that policies associated with known fraudulent activity are terminated, we ignore the PolicyNumber attribute as well.

Month	RepNumber
WeekOfMonth	Deductible
DayOfWeek	DriverRating
Make	DaysPolicyAccident
AccidentArea	DaysPolicyClaim
DayOfWeekClaimed	PastNumberOfClaims
MonthClaimed	AgeOfVehicle
WeekOfMonthClaimed	AgeOfPolicyHolder
Sex	PoliceReportFiled
MaritalStatus	WitnessPresent
Age	AgentType
Fault	NumberOfSuppliments
PolicyType	AddressChangeClaim
VehicleCategory	NumberOfCars
VehiclePrice	~~Year~~
FraudFound	BasePolicy
~~PolicyNumber~~	

Table 3. Original Auto Insurance Fraud Features

Values in the data set have been pre-discretized (probably for anonymization); e.g. the distribution of VehiclePrice appears in Table 4.

Value	Frequency
Less than 20,000	1,096
20,000 to 29,000	8,079
30,000 to 39,000	3,533
40,000 to 59,000	461
60,000 to 69,000	87
More than 69,000	2,164

Table 4. VehiclePrice Distribution

We constructed Random Forest classification models for both the original features and the reputation features, as described in section III. To be consistent with previously reported results, claims from 1994 and 1995 were used as training data, and claims from 1996 were used as testing data. The primary evaluation measure is cost savings, as this was reported in earlier publications and it directly relates to the core goal for a fraud detection system: cost reduction. Table 5 shows an example of a confusion matrix describing the following counts:

- True Positives (TP): the number of claims in the test set that are predicted to be Fraud and are actually Fraud
- False Positives (FP): the number of claims in the test set that are predicted to be Fraud but are actually Not Fraud
- False Negatives (FN): the number of claims in the test set that are predicted to be Not Fraud but are actually Fraud
- True Negatives (TN): the number of claims in the test set that are predicted to be Not Fraud and are actually Not Fraud

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	TP	FN
	Not Fraud	FP	TN

Table 5. Example of Confusion Matrix

Given classification results, as shown in Table 5, costs can be computed as follows:

$$\begin{aligned} &TotalCost = \\ &InvestigationCost * TP \\ &+ \left(InvestigationCost + ClaimCost \right) * FP \\ &+ ClaimCost * \left(FN + TN \right) \end{aligned} \tag{11}$$

In [19], the average InvestigationCost was given as \$203 and the average ClaimCost was given as \$2,640. We use these values as well for consistency. We ran 10 trials for both the Original Features (OrigF) approach and the Reputation Features (RepF) approach.

Using the Original Features (OrigF), we constructed 10 Random Forests from the 11,337 original feature vectors from 1994 and 1995. Each of these 10 Random Forests was evaluated on the 4,083 original feature vectors from 1996.

Using the Reputation Features (RepF), we partitioned the 5,195 reputation feature vectors from 1995 into two subsets (as we used 12 months of history to construct reputation features). For 10 iterations, the first subset was used to construct our one-class SVMs and the second subset was used to construct a Random Forest classifier. Each of the 10 Random Forests was then evaluated on the 4,083 reputation feature vectors from 1996.

Balanced stratified random sampling was used for constructing Random Forests for both the original features and the reputation features. A total of 2,000 trees were constructed for each Random Forest model, with the ceiling of the square root of the number of input features used as the number of randomly selected features to consider for each decision node. For both Original Features (OrigF) and Reputation Features (RepF) the Out Of Bag (OOB) estimate of error from the training data [6] was used to select the classification threshold which minimizes cost.

VI. Experimental Results

Cost savings is used as our primary metric of interest. In [19], cost savings was recorded as the difference between the

cost of paying all claims and the cost of using a fraud detection model (Equation 11). In addition to cost savings, we also report the following metrics:

1. Area Under the Receiver Operating Characteristic (ROC) Curve (AUC): the probability that a randomly selected claim from the fraud class will be viewed as more likely to be a fraudulent claim than a randomly selected claim from the not-fraud class
2. Precision: the probability that a predicted fraudulent claim is actually a fraudulent claim (TP/(TP+FP))
3. Recall: the probability that an actual fraudulent claim is predicted to be a fraudulent claim (TP/(TP+FN))
4. F Measure: the harmonic mean of Precision and Recall (2/(1/Precision + 1/Recall))

Table 6 shows evaluation metrics for our experiments. The values for the Reputation Features approach are marked as RepF, while the values for the Original Features approach are marked as OrigF. Standard Error (SD) values are listed to assess statistical significance.

	RepF	RepF SD	OrigF	OrigF SD
Cost Savings	$189,651	$2,665	$165,808	$748
AUC	82.0%	0.1%	73.8%	< 0.1%
Precision	13.3%	0.1%	11.2%	< 0.1%
Recall	80.3%	1.1%	94.2%	0.1%
F Measure	22.8%	0.1%	20.0%	0.0%

Table 6. Experimental Results

Table 7 shows the average confusion matrix for the Reputation Features approach.

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	171.0	42.0
	Not Fraud	1,118.6	2,751.4

Table 7. Average RepF Confusion Matrix

Table 8 shows the average confusion matrix for the Original Features approach.

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	200.6	12.4
	Not Fraud	1,591.4	2,278.6

Table 8. Average OrigF Confusion Matrix

Figure 3 compares the Receiver Operating Characteristic (ROC) curves for the two approaches to fraud detection.

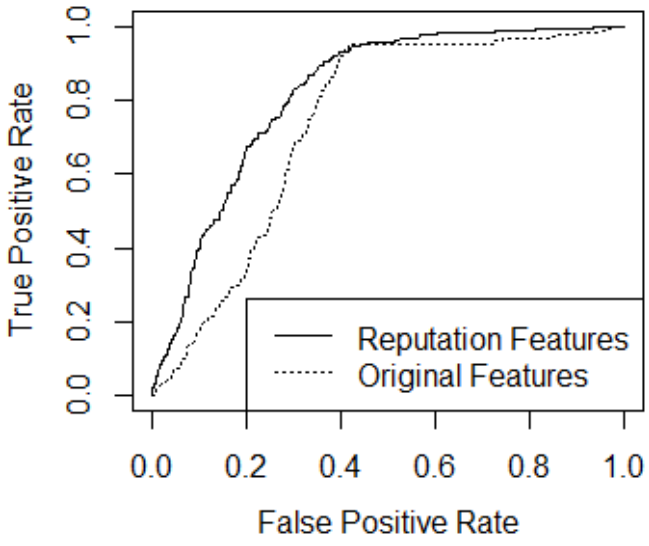

Figure 3. ROC Curves

Figure 4 identifies the most important classification features for the Reputation Features approach. Both the one-class SVM similarity feature for the Fraud class and the Legitimate class are identified as important features.

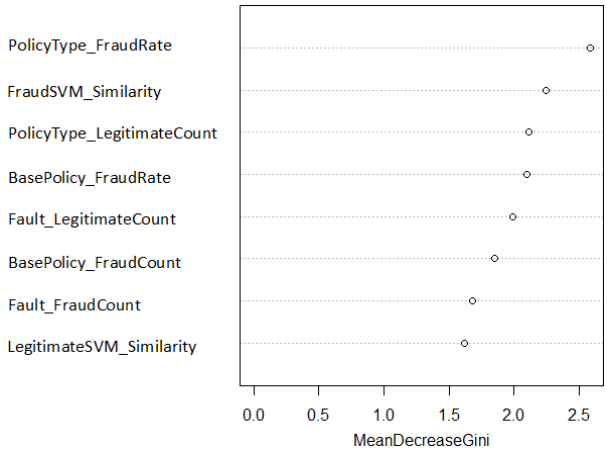

Figure 4. Most Important Reputation Features

As shown in Table 6, the Random Forest classifier constructed from the Original Features is competitive with previously reported state-of-the-art results. The previously reported state-of-the-art results for cost savings was $167,000, while the upper bound of the 95% confidence interval for the Original Features approach shown in Table 6 is $165,808 + 1.96 * 748 = $167,274. The cost savings for the Reputation Features approach is 13.6% higher than the previously reported state-of-the-art results:

($189,651 - $167,000) / $167,000 = 13.6%

It's interesting to note that the operating threshold for the Original Features approach occurs where the two ROC curves meet; but the operating threshold for the Reputation Features approach occurs in the region where the False

Positive rate is 10% lower. Though the True Positive rate (recall) is lower for the Reputation Features approach, the overall cost is significantly reduced.

VII. CONCLUSIONS

The use of deception for financial gain is a commonly encountered form of fraud. Costs for the affected companies are high, and these costs are passed on to their customers. Detection of fraudulent activity is thus critical to control these costs. We proposed to address insurance fraud detection via the use of reputation and similarity features that characterize insurance claims and ensemble learning to compensate for changes in the underlying data distribution. A publicly available auto insurance fraud data set was used to evaluate our approach. Our approach showed a 13.6% increase in cost savings compared to previously published state of the art results for the auto insurance fraud data set. Though an auto insurance fraud data set was used for this demonstration, reputation features could easily be applied to other fraud detection domains, including health care insurance fraud, credit card fraud, securities fraud, and accounting fraud. This approach could also be useful for other applications, such as credit risk classification [23] or computer network intrusion detection [24].

Future extensions include investigation into the use of alternative reputation history lengths. For example, we will explore the use of reputation features based on the most recent 3, 6 and 9 month intervals (in addition to the existing 12 month interval). We also plan to investigate the utility of updating the one-class SVMs on a monthly basis, and synthesizing data to show robustness against adversarial changes to the underlying data distribution for the fraud class.

REFERENCES

[1] B.T. Adler, L. de Alfaro, S.M. Mola-Velasco, P. Rosso, and A.G. West. "Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features", Proceedings of the 12th Intl Conf on Intelligent Text Processing and Computational Linguistics, 277-288, 2011.

[2] Association of Certified Fraud Examiners Report to the Nations, http://www.acfe.com/rttn.aspx, last accessed 2013-03-18.

[3] Auto Insurance Fraud Data, http://clifton.phua.googlepages.com/minority-report-data.zip, last accessed 2013-03-18.

[4] R.A. Becker, C. Volinsky, and A.R. Wilks. "Fraud Detection in Telecommunications: History and Lessons Learned", Technometrics, 52(1), 20-33, 2010.

[5] R.J. Bolton and D.J. Hand "Statistical Fraud Detection: A Review", Statistical Science, 17(3), 235-255, 2002.

[6] L. Breiman. "Random Forests", Machine Learning, 45(1), 5-32, 2001.

[7] B. Caputo, K. Sim, F. Furesjo, and A. Smola. "Appearance-Based Object Recognition Using SVMs: Which Kernel Should I Use?", Proceedings of the Neural Information Processing Systems Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision, Whistler, 2002.

[8] N.V. Chawla, K.W. Boyer, L.O. Hall, and W.P. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research, 16, 321-357, 2002.

[9] P.K. Chan and S.J. Stolfo. "Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection", Proceedings of the 4th Intl Conf on Knowledge Discovery and Data Mining, 164-168, 1998.

[10] C. Chen, A. Liaw, and L.Breiman. "Using Random Forest to Learn Imbalanced Data", University of California at Berkeley Tech Report 666, 2004.

[11] D. DeBarr and Z. Eyler-Walker. "Closing the Gap: Automated Screening of Tax Returns to Identify Egregious Tax Shelters", SIGKDD Explorations, 8(1), 11-16, 2006.

[12] P. Domingos. "MetaCost: a General Method for Making Classifiers Cost-Sensitive", Proceedings of the 5th Intl Conf on Knowledge Discovery and Data Mining, 155-164, 1999.

[13] R.O. Duda, P.E. Hart, D.G. Stork. "Bayesian Decision Theory", Pattern Classification, 2nd ed,Wiley & Sons, 20-83, 2001.

[14] T. Fawcett and F. Provost. "Adaptive Fraud Detection", Data Mining and Knowledge Discovery, 1(3), 291-316, 1997.

[15] Federal Bureau of Investigation Financial Crimes Report to the Public, Fiscal Years 2010-2011, http://www.fbi.gov/stats-services/publications/financial-crimes-report-2010-2011, last accessed 2013-03-18.

[16] A. Gepp, J.H. Wilson, K. Kumar, and S. Bhattacharya. "A Comparative Analysis of Decision Trees Vis-a-vis Other Computational Data Mining Techniques in Auto Insurance Fraud Detection", Journal of Data Science, 10, 537-561, 2012.

[17] H. He and E.A. Garcia. "Learning from Imbalanced Data", IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284, 2009.

[18] J. Li, K.Y. Huang, J. Jin, and J. Shi. "A Survey on Statistical Methods for Health Care Fraud Detection", Health Care Management Science, 11(3), 275-287, 2008.

[19] C. Phua, D. Alahakoon, and V. Lee. "Minority Report in Fraud Detection: Classification of Skewed Data", SIGKDD Explorations, 6(1), 50-59, 2004.

[20] B. Scholkopf, J.C. Platt, J. Shawe-Taylor, A.J.Smola, and R.C. Williamson. "Estimating the Support of a High-Dimensional Distribution", Microsoft Technical Report MSR-TR-99-87, 1999.

[21] S. Viaene, R.A. Derrig, and G. Dedene. "A Case Study of Applying Boosting Naive Bayes to Claim Fraud Diagnosis", IEEE Transactions on Knowledge and Data Engineering, 16(5), 612-620, 2004.

[22] E.B. Wilson. "Probable Inference, the Law of Succession, and Statistical Inference". Journal of the American Statistical Association, 22, 209-212, 1927.

[23] K. Bache and M. Lichman. "Statlog German Credit Risk Data Set", UCI Machine Learning Repository, http://archive.ics.uci.edu/ml /datasets/Statlog+%28German+Credit+Data%29, last access 2013-05-24.

[24] "KDD Cup 1999 Computer Network Intrusion Detection Data set, http://www.sigkdd.org/kdd-cup-1999-computer-network-intrusion-detection, last accessed 2013-05-24.