Provisional Research on Ensemble Learning Techniques for Card Fraud Detection

Pooja Pant, Prakash Srivastava, Ashutosh Gupta

ABSTRACT--- Machine learning have revolutionized fraud detection in various domains like telecommunication and e-commerce. Global statistics shows how billions of dollars are lost because of card frauds every year and millions of people falling the victims. Fraud detection systems used for credit card fraud detection 2 decades ago are still being used because of the trust and stability they have provided for so long. With a number of academic research being done in fraud detection their effect on the financial industry has been minimum. Even with high prediction accuracy using machine learning approaches like deep learning and stack ensemble most of these research gets directly rejected by the industry. Our research objective is to highlight the reason of rejectection which are mostly ignored by the researchers and there adverse effect on the results

Keywords: Ensemble Learning , Machine Learning ,Fraud Detection

I.INTRODUCTION

Financial domain is considered "a profit target" as fraudsters can make a lot of money in a very short span of time with less risk. Many a time a fraud gets reported days after it has happen. Millions of transactions take place on daily basis with an approximation of 1% fraud around the globe. This small percent has huge impact on the financial institutions not only in terms of the money lost but also the inconvenience caused to its customers**.** Financial institutions continuously seek better methods for prevention and detection of frauds. Machine Learning has revolutionized fraud detection segments for telecommunication and other domain.

Expert systems have been the core of all fraud detection anomaly systems around the globe. Many machine learning algorithms like decision tree and linear regression algorithms are the widely researched algorithms for fraud detection which have been considered by financial institutions for real world fraud detection. Even with better performance as compared to expert systems, machine learning based fraud detection are still kept low priority. The main reason of this mistrust is the cost attached with every transaction and the need of remodeling after a few weeks in the real world. In real world transactions vary adversely with a number of external factors like sale, festivals, trends and volatility playing a major role. Expert systems are developed by the domain experts who have experience in the market and know trend movement , their rules can detect fraud even with the presence trends or seasonality whereas this is not the case for machine learning. Remodeling is a necessary in machine learning to avoid unstable behavior of the model, by the time the researcher get aware of the situation direct loss of money has already taken place.

In this article multiple machine learning and ensemble learning methods are evaluated and analyzed in context with credit card fraud detection with the objective of finding a learning technique which provides better performance in terms of less false cases even in the presence of trends. For this article a real world electronic transaction data set was used. The data set comprises of electronic transactions from ATM , POS machines and E-commerce transactions from the issuer bank working. Further information regarding this dataset will not be disclosed due to security reasons. The outcome of this research is being used to design a state of art Fraud detection system. Our research also tries to highlight the common mistake researchers made by researchers by neglecting the time duration of the data set on which the model is trained, difference in the complexity of linear , ensemble , deep learning models and the section of performance measure instead of relying on just accuracy for a cost sensitive usecase.

This article has been divided into the following sections: section 2 discuss the research methodology adopted for our research. Section 3 discusses ensemble learning and deep learning techniques which are been highly researched upon in finance domain for building automated machine learning models.

In section 4 provides the research experiment in details followed with a conclusion in section 5

II. RESEARCH METHODOLOGY

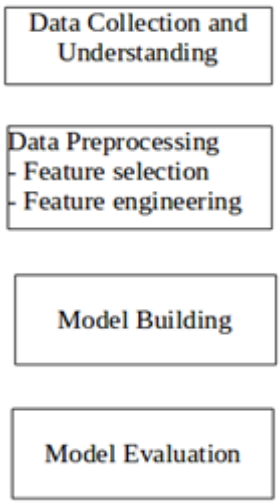
We followed cross industry standard process for data mining(CRISP-DM).This method divides the entire research process into 4 broad categories as shown in figure 1. In the first phase we collected the dataset from a reliable source and gathered domain knowledge to understand the data. In the second phase we extracted the data in desired format, feature selection and feature generation were carried out on the data set. In the third phase we used to train the model using training data and tested the model using test data. Based on prior research we selected machine learning algorithms like ANN, decision tree, logistic regression, KNN, Naive bayes, CART, random forest. The models were evaluated in the fourth phase based on various accuracy and error parameters.

Revised Manuscript Received on August 14, 2019.

Pooja Pant, Amity University, Noida, U.P, India. (Email: ashima81k@gmail.com).

Dr. Prakash Srivastava, Amity University, Noida, U.P, India. (Email: **psrivastava9@amity.edu**).

Dr. Ashutosh Gupta, UPRTOU, Allahabad, U.P ,India. (Email: ashutosh3333@gmail.com).



III. MACHINE LEARNING TECHNIQUES

A number of statistical and machine learning techniques have been used in developing anomaly systems. As discussed earlier parametric machine learning like decision tree and linear regression have always been used for fraud detection.Recently there appeared to be a shift in the focus to non parametric , hyper parametric and more complex models for fraud detection because of the less number misclassification cases raised by them. In this section we will discuss some of the machine learning techniques which have recently become the base of many fraud detection systems.

3.1 Ensemble learning

Ensemble learning are meta algorithms that uses more than one machine learning algorithm as different modules in order to learn from a collection of predictors and provide better results. Ensemble learning is widely used as it increases the accuracy by reducing the error rates ,providing higher consistency by avoiding over fitting and reducing bias and variance errors, therefore increasing the prediction accuracy. There are various ways of getting the final output :the basic way is by model voting where the result with majority is taken into consideration, the second is using weights in this some weights or priorities are assigned to each module and the module with higher priority is preference another way is using predictors as predictors where the output of these modules is considered as the input of another module. The basic ensemble techniques can be classified as bagging , boosting and stacking.

In bagging every learner learns from a subset of the training data which is selected at random with replacement. After the models have learned the same testing data sample is given an input to each learner ,the output of each learner is collected and using various ensembles as explained earlier we get the final result. Random forest is such bagging technique which trains various models using hyper parameters and the final output is collected using voting approach or averaging .Random forest can be used for regression and classification predictions. Random forest comprises of a number of decision trees predicting independently , the output of each of these sub trees is collected and using majority voting a result is deduced. Fussing of multiple trees provides high prediction accuracy. However this technique however provide unsatisfactory results for data with redundant features and high noise, which is caused because of the error rates in the sub-decision trees as errors in decision trees can be minimized on fussing the results together but cannot be removed .[17] proposed an advance decision tree using Bootstrap data split in order to increase the difference between the sub-tree and provides less probability weight to the weak learners . Using In our experiment we have used distributed random forest as bagging ensemble learner.

Algorithm 1 Bagging Algorithm

Input : Training data D = {(X1,Y1),(X2,Y2),(X3,Y3)....(Xm,Ym)}={Xi , Yi }m i=1

Initialization:

Base Learning classifier -> F

Number of learning rounds -> T

Process :

For : t <- 1 to T:

Generate a bootstrap: Dt = Bootstrap(D)

For every sample of D:

Train base classifier ft from Dt

ft=F(Dt)

Return: F(x) =argmaxy€Y ∑ft(x)=$_y$

Boosting or adaptive boosting is an iterative algorithm which uses concept similar to bagging except, the learners with low performance are remodelled in order to increase their performance[28] . In ad boost a learner learns using a sample from training data the but in boosting the testing of the learner is done using the entire training dataset. The significant errors reported are then weighted, for the second learner a subset is randomly chosen from the training dataset but the instances with significant errors are given higher priorities. All the learners are then tested using the entire training data set and the significant error is calculated . This process continues until the number of learners equals to the predefined number in the algorithms. The pseudo code of adaptive boosting algorithm is shown in algorithm 2. In our experiment we used gradient boosting machine which aims at building strong learners using weak learners.

Algorithm 2 AdaBoost Algorithm

Input : Dataset D = {(X_1,Y_1),(X_2,Y_2),(X_3,Y_3)....(X_m,Y_m)}={Xi , Yi }^{m} $_{i=1}$

Initialization: Base Learning classifier -> F

Number of learning rounds -> T

The weight distribution : D(i)= 1/n

Normalization facto Z_t

Process :

For : t <- 1 to T:

Step 1 : Train the base classifier F_t from D using distribution D_t

f_t=F(D,D_t)

Step 2 : Measure error rate ε of f_t

$ε_t$ =∑$_{ft(xi)≠yi}$$D_t$(i)

Step 3: Measure weight a_t of f_t

a_t = 1/2 ln [(1-z_t)/z_t]

Step 3 :Update weight

$$D_{t+2}(i) = [D_t(i)e^{-a(yifi(xi))}]/Z_t$$

Return: $F(x) = sign(\sum a_t f_t(x)) : t= 1,2...T$

Stacking is an ensemble learning where a diverse set of learners or algorithms are used for better prediction by observing the best combination of the learners on the same training dataset. Stack ensemble is a multi-stack approach , where in the first stack different learners are trained and in the second approach the best combination of learners is considered using machine learning. The second approach can be termed as meta learning as an algorithm is to be decided for combining the results of the independent learners. Stacking uses the concept of k-fold cross validation in the first approach will provide a result column for the values predicted and these can be used for checking the accuracy of the learners . A matrix is formed at the end of the first approach on which meta learning is to be performed .In our experiment we performed stacking using random forest , gradient boosting machine and deep learning.

Algorithm 3 Stacking

Input : Dataset $D = \{(X_1,Y_1),(X_2,Y_2),(X_3,Y_3)....(X_m,Y_m)\}=\{Xi , Yi\}^m_{i=1}$

Initialize : $x'_i = \{h_1(x_i),h_2(x_i).....h_r(x_i)\}$

H <-An ensemble classifier

Process :

Step 1 learn from first level classifiers

For t <- 1 to T

learn base classifier h_i ***based on D***

Step 2 Construct new data set from D

For i <- 1 to m

Construct a new dataset $\{x'_i,y_i\}$

Step 3 learn from second level classifier

learn a new classifier h' based on $\{ x'_i,y_i \}$

Output: $H(x) = h'[h_1(x) , h_2(x) .....h_r(x)]$

3.2 Deep learning

Deep learning can be considered a subcategory of neural network which works on the concept multilayered neural network. In which the model is build using gradient decent calculated from back propagation neural network. Deep learning can be done using the concepts of auto-encoders and Restricted Boltzmann Machines[25][11]. At each hidden layer level the neurons contain hyperbolic tangent functions and max out activation function. More complex and theoretically advanced concepts like Adaptive learning rate, Point of detection and grid search can used in order to get high predictability. The working steps of neural network are as follow .

Step 1 : assign random weights to all the linkages

Step 2 :find the activation rate of hidden nodes using the inputs and the linkages

Step 3: find the activation of output nodes using step 2

Step 4: Find the error rates at the output nodes

Recalibrate all the linkages between hidden nodes and output nodes

Step 5: Cascade down the error using using step 1 and step 4

Step 6 : Recalibrate the weights between hidden nodes and the input nodes

Step 7 Repeat the process until convergence criteria is met

Step 8 :Score the activation weight using final linkage weights

IV. EXPERIMENTAL SETUP & RESULTS

The objective of the experiment is to evaluate the performance of different algorithm like neural network , bagging and boosting individually, evaluate the effect of conceptual drift and provide the reasoning for using error estimation methods instead of just accuracy .

4.1 Dataset

For our research experiment we used a real world dataset with 200K electronic transactions and a fraud ratio of 2:100.In our first data we used the original data set and labelled it D1 and created a subset D2 was extracted on the basis of seasonal trends. Some features of the driven data are shown in table 1.For security issue any information regarding the financial institution or a detailed description of the data set cannot be disclosed. The real world data used for our experiment was used for creating features. Some of driven features are shown in the table.

#	Table	Values
1	Msg_type	{0,1,2}
2	Trxn_Origin	{1,2,3}
3	Trxn_Type	{E,A,P,M}
4	CP	{0,1}
5	CNP	{0,1}
6	Issuer_Currency	{356,840}
7	Mcc_type	{....}
8	Mcc_category	{....}
9	Cross-Country	{0,1}
10	Service_code	{2,3}
11	HighRisk_Mcc	{0,1}
12	*Timespread*	{1,2,3,4}
13	SeasonSale	{0,1}
14	Amount_Bucket	{L,M,H,}
15	Dormant_3week	{0,1}
16	last_recent_region	{0,1}
17	HighRisk_Merchant	{0,1}
	.....	

Table 1 Dataset description

4.2 Experimental setup

In the first experiment we choose multiple machine learning algorithms like decision trees , random forest , neural network , Naive Bayes , KNN etc,. commonly used for fraud based research. The models were trained on both the dataset D1 and D2. The performance of the models were evaluated based on the accuracy and the presence of conceptual drift. The best performance models were selected and modeled individually and by using stack ensemble in our second experiment . For our second experiment the performance was based on the misclassification.

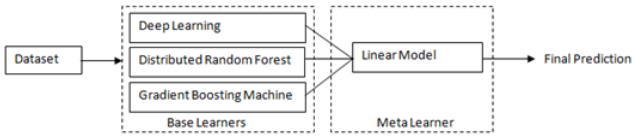


Figure 2 Experiment 2 Stack ensemble setup

4.3 Performance Measures

MSE also known as mean standard deviation is the mean of the square error rate or distance between the prediction and the actual value.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Root Mean Square Error(RMSE)

RMSE quadratic scoring rule which can be used for measuring the average error rate or distance between the prediction and the actual continous values.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Where:

- N is the total number of rows (observations) of your corresponding dataframe.
- y is the actual target value.
- $\hat{y}$ is the predicted target value.

LogLoss

Logloss is used for measuring the performance of a biomial or multinomial classification model.

- *N* is the total number of rows (observations) of your corresponding dataframe.
- *W* is the per row user-defined weight (defaults is 1).
- *C* is the total number of classes (C=2 for binary classification).
- *p* is the predicted value (uncalibrated probability) assigned to a given row (observation).
- *y* is the actual target value.

Binary classification equation:

$$Logloss = -\frac{1}{N}\sum_{i=1}^{N} w_i(y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i))$$

Multiclass classification equation:

$$Logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} w_i(y_{i,j} \ln(p_{i,j}))$$

Area Under the Curve

AUC is used to measure the performance on the basis of classifier capability of distinguishing between true positives and false positive. Further the area tend to 1 the better the classifier.

Precision and Recall

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Let me put in the confusion matrix and its parts here.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

F1-Score is the weighted average of Precision and Recall. The higher the F1-Score, the better the model. For all three metric, 0 is the worst while 1 is the best.

$$F1 = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

MCC

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

4.4 Results

In the first experiment the adverse effect on the models due to the presence of conceptual drift can be observed . Models with higher complexity like neural network , random forest were observed to provide higher accuracy even with the presence of conceptual drift whereas algorithm like naive bayes and SVM radial showed dramatic difference as seen in table 2. In experiment 2 we extracted the trend data and made a new data set D3 which was used to train the selected models individually and then stack the algorithms to evaluate the performance of individual models and stacked model. It was observed that random forest provided higher MCC score as compared to neural network

and ensemble stack. But the mean square error and the logloss was observed to be minimum in the case of stack ensemble.

Table 2 result of experiment 1

Algorithm / Dataset	Logistic Regression	SVM Radial	KNN	Naive Bayes	Neural Network	CART	C5.0	Bagged CART	Random Forest	Stochastic Gradient Boosting
D1	0.795652	0.67826	0.74347	0.77391	0.81739	0.76086	0.79130	0.78695	0.8	0.78695
D2	0.95192	0.99799	0.94608	0.97080	0.9988	0.93253	0.96856	0.98185	0.99583	0.97954

Training Dataset K-fold cross validation results

Algorithm	MSE	RMSE	LogLoss	AUC	gini	MCC
Deep Learning	0.000327307725013 66357	0.0180916479352 67356	0.00426991182941 3284	0.861917772212 0596	0.723835544424 1191	0.8129783
Random Forest	0.000708154743607 7082	0.026611177	0.00304352271765 58994	0.992647891628 058	0.985295783256 116	0.840647705
Gradient Boosting Machine	0.000865040265728 7884	0.0294115668696 65215	0.00364600170864 5998	0.999949913878 5838	0.999899827757 1676	0.29481144 220
Stack Ensembling	0.000253616646344 99913	0.0159253460353 29944	0.00115961966180 12095	0.999931081496 9312	0.999862162993 8624	0.9077661.1 239

Testing Dataset results

Algorithm	MSE	RMSE	LogLoss	AUC	gini	MCC
DeepLearning	0.000327140810 19873857	0.01808703431 1869334	0.00675207681 2016536	0.856875998 6474749	0.713751997 2949498	0.741087805
Random Forest	0.000491973407 0596835	0.02218047355 354893	0.00207147857 41630802	0.999071796 5378011	0.998143593 0756022	0.925777138
Gradient Boosting Machine	0.000632515016 5353657	0.02514985122 292706	0.00335128219 8866876	0.963417181 0460853	0.926834362 0921706	0.66691641
Stack Ensembling	0.000282508948 20070773	0.01680800250 4780505	0.00147188695 29107654	0.999012126 3152312	0.998024252 6304623	0.777502

V. CONCLUSION

A number of academic research has been done in financial domain but always get restricted to academics because the unseen dimensionality like conceptual drifts, imbalance datasets and the evaluation measures. This our research tried to highlight that ignoring the conceptual drift in the data shows adverse effect on nearly all the algorithms which are being used for fraud detection. Random forest and neural network outperformed from the rest of the models even with the presence of conceptual drift but the effect was still observed. Performance measure like accuracy have been considered for balanced dataset and not for imbalance data where the difference in class distribution is large. But our research shows that accuracy can only provide a general even for a balanced dataset. From the experiment we could state that stack ensemble methods provided a better performance with respect of conceptual drift performance. When the evaluation is based on false classification and error rates it was observed that stack ensemble outperformed in terms of mean square error and logloss. But the cost of correlation was observed to be best in case of random forest. From this research we can also state that ensemble learning methods can outperform complex computational models like neural in terms of stability and misclassification rates.

REFERENCES

1. Ahmad, A., & Safaria, T. (2013). Effects of Self-Efficacy on Students' Academic Performance. *Journal of Educational, Health and Community Psychology*, 2(1). Retrieved from https://www.researchgate.net/publication/263162945_Effects_of_Self-Efficacy_on_Students'_Academic_Performance.
2. Alci, B. (2015). The Influence of Self-Efficacy and Motivational Factors on Academic Performance on General Chemistry Course: A Modelling Study. *Academic Journals*, 10(4), 453-461. doi:10.5987/ERR2014.2003
3. Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191-215. doi:10.1037/0033-295X.84.2.191
4. Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. Englewood Cliffs, NJ: Prentice-Hall.
5. Bandura, A. (1994). Self-efficacy. In V. S. Ramachaudran (Ed.), Encyclopedia of human behavior (Vol. 4, pp. 71-81). New York: Academic Press. (Reprinted in H. Friedman [Ed.], Encyclopedia of mental health. San Diego: Academic Press, 1998).
6. Becker, S., & Gable, R. (2009). The Relationship of Self-Efficacy and GPA, Attendance, and College Student Retention. NERA Conference Proceedings 2009. Retrieved from http://opencommons.uconn.edu/cgi/viewcontent.cgi?article=1003&context=nera_2009
7. Bresó, E., Schaufeli, W., & Salanova, M. (2011). Can a self-efficacy-based intervention decrease burnout, increase engagement, and enhance performance? A quasi-experimental study. Higher Education, 61(4), 339-355. Retrieved from http://0-www.jst or.org.lib1000.dlsu.edu.ph/stable/41477800
8. Burnett, R., Xu, L., & Kennedy, S. (2010). Student Self Efficacy in Intermediate Accounting: A Tool to Improve Performance and Address Accounting Change. The Accounting Educators' Journal, 20, 109-134. Retrieved from www.aejournal.com/ojs/index.php/aej/article/download/167/94
9. Christensen, T. B., Fogarty, T. J., & Wallace, W. A. (2002). The association between the directional accuracy of self-efficacy and accounting course performance. Issues in Accounting Education, 17(1), 1+. Retrieved from o.galegroup.com.lib1000.dlsu.edu.ph/ps/i.do?p=GPS&sw=w&u=dlsu&v=2.1&it=r&id= GALE%7CA83993649&asid=64f6234b8dd2ea0fbaf6af652f3daf0e
10. Clark, S. D., & Latshaw, C. A. (2012, Winter). "Peeling the onion" called student performance: an investigation into the factors affecting student performance in an introductory accounting class. Review of Business, 33(1), 19+. Retrieved from o.galegroup.com.lib1000.dlsu.edu.ph/ps/i.do?p=GPS&sw=w&u=dlsu&v=2.1&it=r&id= GALE%7CA326656943&asid=5838a5f 9f193c7dfcccdab0791d33370
11. Dull, R. B., Schleifer, L. F., & McMillan, J. J. (2015). Achievement Goal Theory: The Relationship of Accounting Students' Goal Orientations with Self-efficacy, Anxiety, and Achievement. Accounting Education, 24(2), 152. doi:10.1080/09639284.2015.1036892
12. Eskew, R., & Faley, R. (1988). Some Determinants of Student Performance in the First College-Level Financial Accounting Course. The Accounting Review, 63(1), 137-147. Retrieved from