Predicting Fraud in Electronic Commerce: Fraud Detection Techniques in E-Commerce

Amitha Raghava-Raju
Societe Generale Global Solution Center
#19, 5th cross, 3rd main, Pampa Extension
Hebbal Kempapura, Bangalore - 560024

ABSTRACT

Electronic commerce is a commercial transaction that involves the process of buying and selling goods, services and transfer of information between businesses and customers over an electronic medium. Electronic commerce has expanded rapidly and continues to grow at an unforeseen rate. With the advent of e-commerce, the potential fraudulent activities are prevalent, and therefore hundreds of millions of dollars are lost to fraud every year. The goal of this paper is to implement and evaluate several anomaly detection methods for making predictions using, or finding patterns in, heterogeneous e-commerce data to detect fraudulent activities of users. Various Machine Learning algorithms – K-nearest neighbors, Random-Forest and Isolated forest algorithms are employed to train the model in order to detect fraud and anomalous techniques in e-commerce.

Keywords

Electronic Commerce, Fraudulent activities, Anomaly detection, K-nearest neighbors, Random-Forest, Isolated Forest.

1. INTRODUCTION

Electronic commerce - e-commerce is a business model that facilitates the transactions of goods, services, funds or data over an electronic network, specifically the internet. E-commerce business transactions can be sub-divided into four categories – Business to Business, Business to Consumer, Consumer to Business, and Consumer to Consumer. E-commerce has allowed firms to establish a market presence, by providing efficient distributed chain for their products and services. The upsurge of e-commerce is accompanied by a drastic increase in Fraud. Fraud detection is applicable to many industries, making fraud detection more important than ever. An important early step in fraud detection is to identify factors that can lead to fraud. What specific phenomenon typically occurs before, during, or after a fraudulent incident? What characteristics are generally seen in fraud? When these aspects, factors and characteristics are pinpointed, predicting and detecting fraud becomes a much more manageable task.

To help find such factors, a dataset from an e-commerce website is used to predict and observe the patterns in data to identify fraudulent activities on ample number of factors and users. It is important to both correctly identify fraudulent behaviour when it arises and to not flag normal user as fraudulent one, thereby alienating customer base and achieving high recall. The approach to evaluate and ascertain what “normal” user behavior is in terms of time spent by the user browsing the website, how fast they move through the sales funnel, their age, gender, time of the year, etc. These characteristics are all represented by numbers and tend to be uniform for a “normal” user. However for a fraudulent user, some of these numbers many deviate from the norm, which must be captured by the model. The model leverages these numbers and advanced statistical techniques to characterize how different fraudulent user behavior is from that of a normal user. Using the fraud detection techniques to examine the dataset from an e-commerce website - a user can be classified as risky if a single device or ip address is correlated to multiple user login profiles or if the duration of time spent from login to purchase occurs in extremely short time periods[1][9].

The main contributions of this paper are as follows. Firstly, build an anomaly detection model that predicts the probability that the transaction of a user is fraudulent. Secondly, validate and authenticate the predictions made by the model, and describe the kind of users likely to be classified as risky users without turning away valid paying customers inadvertently. Lastly, describe the product perspective usage, user experience to be built on the model output. Perhaps further work could provide more features and characteristics to develop policies, monitor orders, and check ip addresses and email addresses to detect fraud and prevent it at an earlier stage.

2. RELATED WORK

According to Fraud Benchmark Report, 83% of North American businesses conduct manual reviews, and on an average, they review 29% of orders manually. Involvement of humans gives insight about fraud patterns and genuine customer behavior. These insights can fine tune automated screening rules. But manual review is costly, time-consuming and leads to high false Negatives [2].

The major disadvantage of the traditional approach is the occurrence of false positives. This means that completely normal customers just looking to make a purchase will be labeled risky and dropped from the business. A false positive not only affects the sale but also lifetime value generated from the customers. Thus manual reviews based on rules should be the last line of defense in the fraud detection strategy.

Recent works include the incorporation of Machine Learning techniques to significantly reduce human effort to identify and gauge the importance of anomalous patterns to detect fraudulent activities in e-commerce transactions. A Popular statistical techniques used previously to build predictive models is the Logistic Regression. The huge dataset consisted of variable number of features. It provided value by asserting predictive power of Individual variables or a combination of variables as a part of a larger fraud strategy. In this technique, the authentic transactions are compared with fraudulent ones to create an algorithm [10].

Another experimental example for anomaly detection is the KDD Cup 99 detection dataset. It consisted of 4.8 million records with 42 features. The dataset consisted of normal data dispersed with attack types. Amazon ec2 instances were used to carry out the experiment as the dataset is very large. Ten models were built using the WSO2 Machine Learner by changing the hyper parameter and the model configuration. The experimental model achieved an accuracy of 95.92% [3].

3. DATASET AND FEATURES

Many e-commerce companies collect and garner good deal of information about their customers through their activities on their website. This data was utilized to make inference from, as it was most available and had ample number of features and very large number of samples. The primary dataset used was that of an e-commerce website that sells hand-made clothes. It contains transitions of many different users as well as provides information on whether the individual's behavior could be categorized as fraudulent or not. An additional dataset was also utilized to correlate the ip-address of the user to the corresponding country.

3.1 Features

The dataset used consisted of 11 features. Five of the features were numerical; the remaining six were categorical features. The features included: userid, signup-time, purchase-time, purchase-value, device-id, source, browser, sex, age, ip-address, country and class.

Not all features provide valuable information to identify fraud. Basic Data Visualization between User ids per device versus Fraud as shown in Figure 1, Sign-up to purchase time versus Fraud as shown in Figure 2 and Fraud Count versus week of the year as shown in Figure 3 provides vital insight in order to detect fraud in the given dataset. Table 1 provides the description of the dataset considered for data modeling.

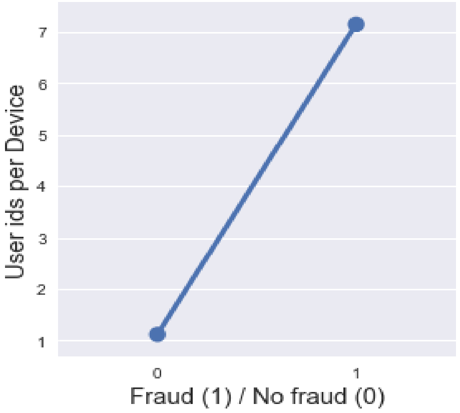

Figure.1 Graph of User ids per Device vs. Fraud

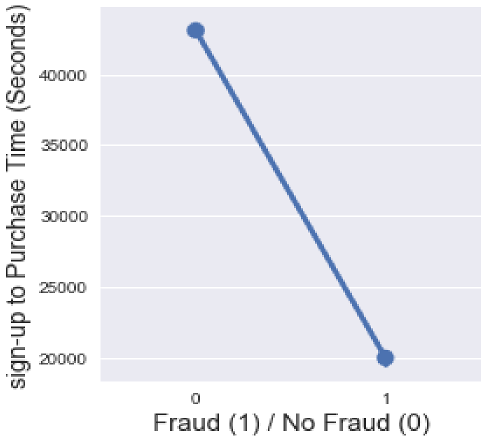

Figure.2: Graph of Sign-up to Purchase Time vs. Fraud

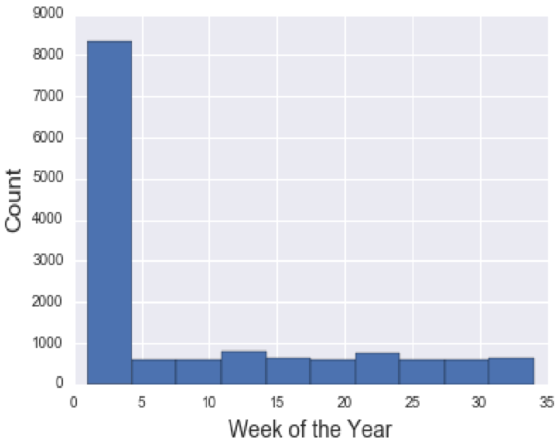

Figure.3: Graph of Fraud vs. Week of the Year

Table 1.Description of the Dataset

	User_Id	Ip_Address	Class
Count	1.511120e+05	1.511120e+05	1.511120e+05
Mean	200171.040970	2.152145e+09	0.093646
Std	115369.285024	1.248497e+09	0.291336
Min	2.000000	5.209350e+04	0.000000
25%	100642.500000	1.085934e+09	0.000000
50%	199958.000000	2.154770e+09	0.000000
75%	300054.000000	3.243258e+09	0.000000
Max	400000.000000	4.294850e+09	1.000000

3.2 Preprocessing

The dataset consists of 151,112 samples, 21,900 samples had missing features. It constituted to 17% of the dataset. Although it was possible to discard the records with missing ip-addresses, but it could be possible that these missing ip addresses correspond to fake addresses generated to perform fraudulent activities. Therefore, it was reasonable to impute these records by labeling the country of origin as "Missing".

Moreover the dataset was evaluated to assess the ratio between the fraudulent samples to normal samples in the dataset. The ratio between fraud to non-fraud samples was determined to be 10:90;

Furthermore, additional columns were introduced into the dataset for efficient analysis of anomalous behavior. These additional columns include userids per device id, signup to purchase in seconds, and country. For anomaly detection algorithms, Gaussian distribution features will benefit the model. Due to the ample number of features in the dataset, feature selection was vital.

The first chosen feature was the userids per device id. Figure 4, depicts the bi-modal relation between the mean userids per device versus the fraud count. The Gaussian distribution is centered around 12. The relation between the signup to purchase in seconds and fraud count was uniformly distributed. Therefore the data was transformed to follow a normal distribution pattern using the transformation equation represented in Eq. (1).

$$\mathrm{X} = \sqrt{2}\,\mathrm{erf}^{-1}(2\Phi - 1) \qquad (1)$$

where X – Normal distribution random variable, Φ – Uniform distribution random variable, and erf^{-1} - Inverse of error function.

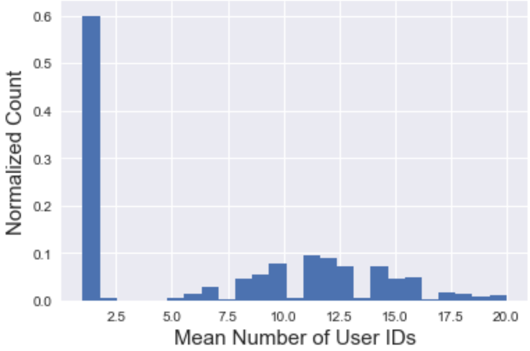

Figure.4: User ids per Device vs. Count

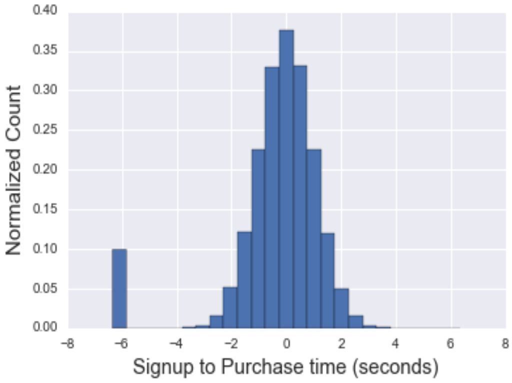

Figure.5: Signup to Purchase time vs. Fraud

The resulting transformed data is normally distributed as shown in Figure 5.Additionally, Feature Selection was performed to select the vital features using the extra trees classifier. Finally from the processed points, the data was randomly split, of which 70% was for training set and the remaining 30% was for testing.

4. METHODS

The goal of this paper is to build an anomaly detection model that predicts the probability that the transaction of a user is fraudulent. Three models were built to detect an anomaly in the dataset using the following machine learning supervised algorithms, K-nearest neighbors algorithm, Random Forest Classifier and Isolation Forest Algorithm.

4.1 K-nearest neighbors algorithm

The KNN algorithm is a robust and versatile classifier. The KNN classifier is also a non parametric and instance-based supervised learning algorithm. Non-parametric means it makes no explicit assumptions about the function or on the underlying data distribution. Instance-based learning means that the algorithm does not explicitly learn the model. Instead, it chooses to memorize the training instances which are subsequently used in the training phase. The K-nearest neighbor algorithm essentially boils down to forming a majority vote between the K most similar instances to a given observation. The similarity is defined by the distance metrics between the two data points. Common choices for distance metrics are Euclidean, Manhattan, Chebyshev and Hamming distance.

The KNN classifier is employed in the project to calculate a decision boundary which is then used to classify new points. For the project, K (number of nearest neighbors) considered was 5 and the chosen metrics is Euclidean distance expressed in Eq. (2).

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + \cdots + (x_n - x'_n)^2} \qquad (2)$$

where *d* is the similarity metrics and $x_1, \ldots, x_n$ are the unseen observations. For a positive integer K (nearest neighbors), observation *x* and metrics d, KNN classifier employed in the project performs the following steps:

i. It runs through the whole dataset computing d (distance) between *x* and each of the training observations. The K points in the training data that are closets to *x* are in set A. Note K is usually odd to prevent tie situation.

ii. The algorithm then estimates the conditional probability for each class, that is, the fraction of points in A with that given class label. The conditional probability formula to evaluate *x* is given in Eq. (3).

$$\mathrm{P}(y = j \mid \mathrm{X} = x) = \frac{1}{K} \sum_{i \in A} I(y^{(i)} = j) \qquad (3)$$

where *x* is the argument and I(*x*) is the indication function. Note. The indication function, I(*x*) evaluates to 1 when the argument *x* is true and 0 otherwise. Finally, the output *x* gets assigned to the class with the largest probability [4].

4.2 Random Forest Algorithm

The Random forest is a tree-based supervised algorithm which involves building several decision trees, then combining their output to improve generalization ability of the model. The method of combining trees is known as ensemble model. The Random Forest Classifier is implemented in the project to train the model. In classification trees, the output is predicted using the mode of observations in the terminal nodes. The splitting decision implemented in the project is the Gini Index (criterion='gini'). The Gini Index is the measure of node purity. If the Gini index takes a smaller value, it suggests that the node is pure. For a split to take place, the Gini index for a child node should be less than that of the parent node. Figure 6 depicts the implementation of Random forest algorithm on a dataset.

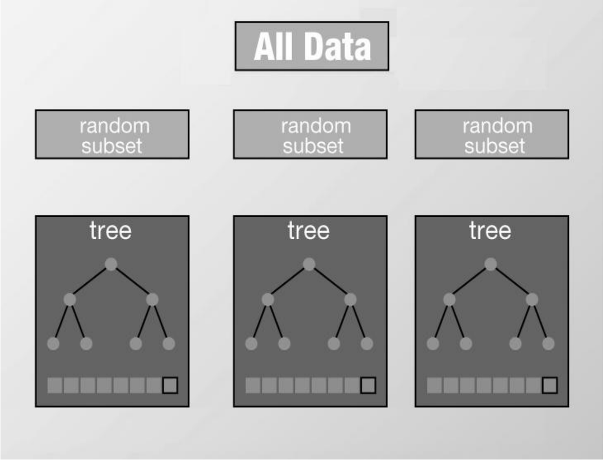

Figure.6: Splitting of data on implementation of Random Forest Algorithm

In the project, the chosen criterion is equated to ‘gini’, bootstrap is set to true, estimators are 10 and minimum sample splits equal to 2. For the mentioned metrics the random forest works in the following way:

i. The Bootstrap Aggregation algorithm creates random samples. Given the dataset D (n rows and p columns), it creates a new dataset d by sampling n cases at random with replacement and 1/3 of the rows are Out of the Bag (OOB) samples.
ii. The model trains d, the OOB samples are used to determine the unbiased estimation of the error.
iii. Out of the p columns in the dataset, $\sqrt{p}$ columns are selected at each node in the data set randomly.
iv. Each tree is grown fully.
v. The final prediction is obtained by averaging or voting when several trees are grown [7].

Finally, the output *x* gets assigned to the class with the largest probability [4].

4.3 Isolation Forest Algorithm

The Isolation forest is a supervised algorithm based on the fact that anomalies are data points that are few and different. As a result of these properties, anomalies are susceptible to a mechanism called isolation. The Isolation Forest or the iForest builds an ensemble of iTrees for a given dataset, and the anomalies are those instances which have shorter average path lengths on the iTrees. There are only two variables in this method: The number of trees to build and the sub-sampling size. The iForest’s detection performance converges quickly with a very small number of trees, and it requires a small accuracy [5].

An anomaly score is required for any anomaly detection method. The difficulty in deriving such a score from the path length h(x) is that while the maximum possible height of iTree grows in order of *n*, the average height grows in the order of log n. The average path length of an iTree is represented by Eq. (4.1).

$$c(n) = 2\mathrm{H}(n - 1) - (2(n - 1) / n) \qquad (4.1)$$

where $c(n)$ is the average path length given n, it is used to normalize path length *h(x)*, H(i) is the harmonic number and it can be estimated by ln(i)+ 0.5772156649 (Euler’s constant). The anomaly score s of the instance *x* is defined in Eq. (4.2).

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \qquad (4.2)$$

where *E(h(x))* is the average of *h(x)* from the collection of isolation trees.

- $E(h(x)) \rightarrow c(n),\ s \rightarrow 0.5$ (4.3)
- $E(h(x)) \rightarrow 0,\ s \rightarrow 1$ (4.4)
- $E(h(x)) \rightarrow n - 1,\ s \rightarrow 0.5$ (4.5)

Eq. (4.3), Eq. (4.4) and Eq. (4.5) depict s is monotonic to *h(x)*. Figure 7. Illustrates the relationship between *E(h(x))* and *s*. Using the anomaly score *s*, the following assessments can be made:

- If instances return *s* very close to 1, then they are definitely anomalies.
- If instances have s much smaller than 0.5, then they are quite safe to be regarded as normal instances.
- If instances all return *s* ~ 0.5, then the entire sample does not really have any distinct anomaly.

In the project a 5-fold cross validation with grid search over a number of estimators is chosen.

The Isolation forest algorithm performs the following steps:

i. The isolation forest isolates observations by randomly selecting features and then recursively selecting random splits values until the sample is isolated.
ii. The recursive splitting can be represented as a tree structure where the number of splitting is equivalent to the path length from the root node to the terminating node.
iii. This path length, averaged over a forest of similarly generated trees is the measure of normality. The shorter the average length, the less work it takes to isolate the sample and more likely it is that the sample is anomalous.

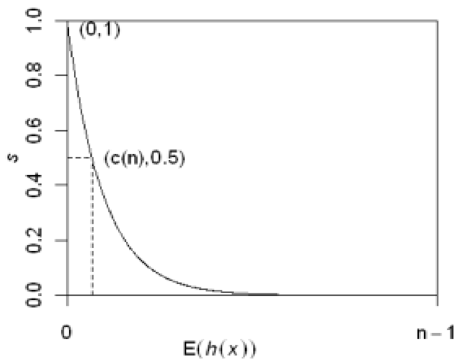

Figure.7: The relationship between expected path length E(h(x)) and anomaly score *s*[6].

5. RESULTS

Brief summary of the results generated by applying the above methods to the E-commerce data set is mentioned in this section. The dataset was split into two portions, where 2/3 of the dataset was used for training and the rest 1/3 of the dataset was used for testing.

The most important features in spotting fraudulent transactions from the given e-commerce dataset were found to be:

i. The speed through which the anomaly traversed from sign-up to purchase.
ii. Number of user ids associated which a device.

Figure.1 shows the relation between the user id to fraudulent behavior. A user who creates multiple login profiles to access a website from the same device cannot be correlated to a normal user behavior. From the dataset, users with multiple login profile from the same device or ip address were flagged as fraudulent users. Figure.2 shows the relation between signup to purchase time versus fraud. It was observed from the dataset, that a normal user spends certain amount of time browsing through the chain of products. On the other hand if the entire sales funnel is traversed in few seconds it indicates fraudulent behavior. Three models were built to detect and predict fraudulent behavior with high precision and to minimize false negatives.

5.1 K-nearest neighbors algorithm

While trying to detect the anomaly using the most vital features from the dataset, the K-nearest neighbors classifier produced correct labels reasonably well. The classifier was able to predict the anomaly with an accuracy of 90.84% and a precision of 0.51 Also the classifier produced 2% false negative, 88% true negatives, 6% false positives and ~3% true positives.

5.2 Random Forest Algorithm

The criterion to train the model for Random Forest Classifier was 'gini', along with 10 estimators. Hence, the splitting decision implemented in the project was based on the Gini Index. The classifier was able to detect the anomaly with an accuracy of 91.23% and a precision of 0.55. This prediction model produced ~ 3% true positives, ~7% false positives, 2% false negatives and 88% true negatives.

5.3 Isolation Forest Algorithm

A 5-fold cross validation with grid search over a number of estimators in the forest was feed to the algorithm to train the model. The algorithm was able to detect anomalies with an accuracy of 84%.The model generated 1% true positives, 8% false positives, 8% false negatives and 82% true negatives; The area under the curve was determined to be 0.53;

The models were constructed to identify fraudulent behavior when it arises and not flag normal users as fraudulent.

6. CONCLUSION

Although the dataset had certain number of features, not all the features contributed equally to identify the fraudulent user behavior on an ecommerce website. From Data Visualization models among many features in the dataset versus Fraud, it was observed the signup to purchase time versus fraud and numbers of userids per device versus fraud were most vital features to categorize fraud and interpret the results and observations appropriately. The signup to purchase time provided the speed at which user moved through the sales traffic, lower the timeframe between signup and purchase, more likely to be categorized as fraudulent behavior. Also existence of multiple userid per device or ip address indicates multiple login profiles of a fraudulent user. Another observation drawn from the model is that the fraudulent user behavior is high during the first few weeks of the year as shown in Figure 3. Three supervised learning algorithms were employed to train the model in order to predict and detect anomaly with great precision and accuracy. First model was the K-nearest neighbors: supervised algorithm analyzed the dataset based on the similarity metrics with respect to its neighbors. The similarity metric employed is the Euclidean distance to classify and detect fraud in the dataset. Second model was the Random Forest: supervised learning algorithm based on building several decision trees and combining them to form the ensemble tree. The splitting is based on the purity of the node and the Gini index in order to identify aberrant data in the dataset. Third model was the Isolation Forest: supervised learning algorithm based on analyzing how many splits on features are necessary to isolate a given sample. The employed three models try to ascertain "normal" user behavior on various features like time span, acceleration through the sales funnel. These statistical characteristics are almost similar for a normal users but it varies largely for a fraudulent users. These models leverage these statistics and accordingly classify the user normal or fraudulent.

7. FUTURE WORK

Electronic commerce is growing at a phenomenal rate but it is also accompanied by the prevalence of fraudulent user behavior and transactions, therefore it is imperative to take primordial steps to prevent or at the best minimize fraud in e-commerce. Fraud detection techniques must be enhanced. Certain cautionary measures to be implemented in order to minimize and eventually prevent fraud would be to i. Ensure every user must sign up with a complete profile and the user can be given a new user id only when the previous one is deprecated, in this manner detection of a fraudulent user can be made more tractable, ii. In order to easily detect the time span between signup to purchase and rightly flag the fraudulent user, it would be advantageous to attract the user with discounts or offers on common products as normal users are mostly likely to skim through these products, iii. Security must be enhanced at the beginning of the year as well as at sale time, when general user activity is high, iv. Other ensemble or unsupervised learning algorithms can be employed to train the model and predict fraudulent behavior with higher accuracy like OneClassSVM or EllipticEnvelope [8].

8. REFERENCES

[1] "U.S. Online Retail Forecast, 2009 to 2014," Forrester Research, Inc., March 2010.

[2] Online Fraud Report, 11th Annual Edition, Cybersource, February 2010.

[3] S´ergio Moro, Paulo Cortez, Paulo Rita , "A Data-Driven Approach to Predict the Success of Bank Telemarketing", Dec 16, 2016.

[4] Kevin Zakka, "A Complete Guide to K-Nearest-Neighbors with Python and R", Kevin Zakka's Blog, July 13, 2016.

[5] Zhiguo Ding , Minrui Fei, "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window", March 13, 2015.

[6] Fei Tony Liu, Kai Ming Ting, , Zhi-Hua Zhou, "Isolation Forest", 2010.

[7] HackerEarth, "Practical tutorial on Random Forest and Parameter tuning in R", 2017.

[8] Sonya Sawtelle, "Anomaly Detection in Scikit-Learn", 2017.

[9] Magic Quadrant for Web Fraud Detection, Gartner, Inc., January 2010.

[10] Priya J Rana, Jwalant Baria, "A Survey on Fraud Detection Techniques in Ecommerce", International Journal of Computer Applications, Volume 113 – No. 14, March 2015.