Invariant Diversity as a Proactive Fraud Detection Mechanism for Online Merchants

Roy Laurens
Department of Computer Science
University of Central Florida
Orlando, USA
rlaurens@knights.ucf.edu

Jusak Jusak
Department of Computer Engineering
Institut Bisnis dan Informatika Stikom Surabaya
Surabaya, Indonesia
jusak@stikom.edu

Cliff C. Zou
Department of Computer Science
University of Central Florida
Orlando, USA
czou@cs.ucf.edu

***Abstract* — Online merchants face difficulties in using existing card fraud detection algorithms, so in this paper we propose a novel *proactive* fraud detection model using what we call *invariant diversity* to reveal patterns among attributes of the devices (computers or smartphones) that are used in conducting the transactions. The model generates a regression function from a diversity index of various attribute combinations, and use it to detect anomalies inherent in certain fraudulent transactions. This approach allows for proactive fraud detection using a relatively small number of unsupervised transactions and is resistant to fraudsters' device obfuscation attempt. We tested our system successfully on real online merchant transactions and it managed to find several instances of previously undetected fraudulent transactions.**

Keywords—Electronic Commerce; credit card fraud; fraud prevention; diversity index

I. Introduction

Today, the advancement of Internet technology coupled with the proliferation of its applications has enabled millions of users to access and use the Internet for various purposes, such as searching, browsing, chatting, video communication, trading and selling. Utilizing the above communications technology, we have witnessed a shift from traditional commerce to modern transactions that involve worldwide virtual markets comprised of customers across nations and cultures. In the current situation, both debit and credit cards are the most dominant method of payment under the Card Not Present (CNP) scheme. (The CNP transactions imply transactions in which cards are not presented to the merchants at the time of purchase, i.e. customers and merchants are not interacting face-to-face.) This enables real time processing, which allows authorization to be obtained immediately on an individual basis and therefore enables an instant transaction process between customers, merchants and card issuers [1].

However, these universal and attractive features of modern online transactions bring direct consequences that make them vulnerable to fraud. An online merchant is simply unable to positively ascertain the cardholder's identity because the transaction is not being conducted face-to-face. A staggering study indicated that 25% of global fraud losses are caused by CNP fraud, and it represents 45% of total U.S. card fraud [2][3]. Furthermore, transaction fraud has the potential to completely shut down the online merchant's business, as card associations will refuse to process transactions from merchants with significant fraud [4].

Online fraud is especially problematic for US-based merchants. In the United States, e-commerce lacks verification infrastructure such as two-factor authentication that has been widely deployed in many other countries [5][6]. Therefore, the merchant in the US has to bear all of the liability of fraudulent transactions.

Unfortunately, most research towards payment card fraud is focused on the card issuers or payment processors, and has overlooked the merchant side, especially online merchants. These merchants have to resort to simplistic rule-based classifiers with static parameters that are based on previous fraudulent transactions. However, fraudsters can easily obfuscate their data to defeat these conventional detection methods that rely on information such as IP address, geolocation, email address, etc. Additionally, these rules are either based on the merchant's own past fraud patterns (and therefore cannot proactively detect new or emerging fraud patterns) or based on commonly known industry rules (such as high-risk country lists, etc.), which are seldom appropriate for all merchants and can also be easily circumvented by knowledgeable fraudsters.

Furthermore, a time lag of 30 days or more exists between the time a fraudulent transaction occurs and the time when fraud is reported by the cardholder, which means the merchant could suffer from more than a month of undetected fraud. Also, for each buyer/account, a merchant can only observe a small fraction of the person's transactions that are conducted. The card issuer and processor, on the other hand, are able to detect patterns over a more complete and larger transaction volume, which makes it easier for them to detect fraud [7]. Unfortunately, the reality is that card issuers and acquirers are more reluctant to decline a transaction or mark it as fraudulent because they do not want to cause inconvenience to their customers. This puts a merchant somewhat at a disadvantage: it is liable for fraud yet its detection capability is much more limited compared to card issuers and acquirers.

In this paper, we present an innovative fraud detection model that can be implemented by online merchants. It leverages a merchant's unique strength as front-end entity that directly interacts with the buyer's electronic purchasing device (a computer or a smartphone) and combines it with a statistical concept called *diversity index* [19]. The underlying assumption is that any non-unique device attribute must show a diversity of other unrelated non-unique attributes. For example, many devices will have the same 'OS Version', but it is expected that

This work is supported by the National Science Foundation under grant SaTC-EDU-1723587.

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

devices with the same 'OS Version' will exhibit the variation or diversity of Internet Service Providers (ISPs). An extreme lack of diversity might indicate obfuscation attempt by a potential fraudster. This proposed approach has several main advantages for online merchants:

- It detects deviation from norm, therefore it does not need to be supervised and can *proactively* detect emerging fraud patterns. It does not need known fraudulent transactions to train the system.
- It requires *small dataset*, as diversity pattern across a device's attributes can be exposed with just a few transactions.
- Because it detects diversity of values instead of the value itself, it is *more resistant to obfuscation* attempts by fraudsters.

It should be noted that our system's objective is to address the shortcomings of currently available detection tools for merchants and not to replace them completely. We believe our dynamic and proactive approach is a suitable complement for existing static rule-based classifiers.

The structure of this paper is as follows: In the next section, we provide the current landscape of fraud detection. Next, section III provides formulation of our problem and section IV describes the detail of our solution. Section V shows the implementation of our model and Section VI shows the result. Finally, Section VII concludes the paper.

II. Related Work

A. Fraud Detection Tools

The objective of fraud detection is to successfully identify potentially fraudulent transactions so further analysis (such as manual review or phone call authentication) can be performed on them. In general, fraud detection tools can be classified into supervised and unsupervised methods depending on the availability of training data in the machine learning stage or model construction. The supervised methods rely on a set of training data to teach the predictor to classify normal activities or fraudulent behavior at the transaction level. However, in some cases the information about fraudulent transaction may not be readily available, hence, in such situations we can apply unsupervised methods to detect the abnormal transaction. The unsupervised method is to simply search for transaction which are most divergent from the normal one.

We explain some of the commonly applied algorithms for fraud detection tools in this subsection.

1) *Supervised methods:* In supervised methods, models which will be used for classification are set to undergo learning processes based on a set of examples of either existing fraudulent or non-fraudulent behavior. Hence, in the learning process the fraudulent behaviors have been identified and serve as a baseline. Subsequently, once the models have been created, the supervised algorithms perform fraud detection by analyzing several attributes of transactions, such as the account holder, account number, merchant code, etc. to match with one of the two classes, i.e., the fraudulent or non-fraudulent behavior. Most of adaptive computational intelligent algorithms fall in this category, for example Artificial Neural Network [8][9], Artificial Immune System [10], Genetic Algorithm [11] and self-similarity detection systems [12].

2) *Unsupervised methods:* In contrast to the supervised methods, learning processes in the unsupervised algorithms are done without any examples [13]. This is mainly due to the unavailability of prior knowledge of a particular class in the data set where a certain transaction can be categorized as fraudulent or non-fraudulent behavior. As an example, in the online merchant side that has limited access to the database, retrieving the whole information such as the user's pattern or behavior is almost impossible. Hence, in this situation the unsupervised methods can be utilized. Examples of unsupervised methods including Artificial Neural Networks [8][9] and Hidden Markov Model (HMM) [14].

B. Evaluation of the previous works

There is a major similarity between the two fraud detection approaches discussed previously. Both of them require a large amount of dataset, either of fraudulent/legitimate transactions (for the supervised method) or of normal transactions per user/account (for the unsupervised one). Therefore, they cannot be used effectively by online merchants since they only have a limited number of transactions to analyze. As such, merchants usually only use rule-based detection, which is static and require constant and manual tweaking as new fraud emerges.

More importantly, almost all fraud detection tools do the job by analyzing each transaction and then attempting to catch the fraudulent behavior based on the historical data previously modeled [15]. In the case where the fraudsters obfuscate their identity, the algorithms need to be retrained in order to recognize that new pattern. For card transactions, disputes mostly occur after the cardholders receive their statements, which could be up to 30 days after the transactions. Hence, by the time the merchant was informed of the fraud, the pattern is already old and newer fraud patterns might already have happened.

Therefore, a more proactive system is needed. This system should just require a small amount of dataset and is resistant to device obfuscation, so it can be used as a tool to notify the merchant whenever there is a transaction that significantly deviate from norm. Based on this proactive analysis, the transaction can then be investigated in more detail.

III. Adversary Model

In this paper, the online merchant wants to detect deliberate and systematic transaction fraud conducted by 'professional' fraudsters using many stolen payment cards.

Some specific characteristics of these fraudsters are important for our model:

- They want to monetize a stolen card as soon as possible, prior to the card being reported stolen by the real cardholder. As such, the fraudulent transactions will be performed within a short period of time. This is the basis of the *velocity check* commonly performed by rule-based detection methods [16]. At its core, a velocity check detects if the same person (i.e., same email, same IP address, etc.) made numerous purchases within a short timeframe.

- The more sophisticated fraudsters will attempt to obfuscate their digital fingerprint to defeat velocity detection [17]. However, since they have time constraints due to the need to monetize as soon as possible, their obfuscation cannot be thorough and complete. They might reset/reinstall their OS, change ISPs, install new fonts, use a new screen resolution, update the browser, etc. But it is very unlikely that they will do *all* of these for *every single transaction* as it is very time consuming and will diminish their benefit.
- Fraudster obfuscation attempts will make their transactions appear to come from a new customer. Although merchants are less likely to suspect fraud from old customers, fraudsters cannot make their transactions appear to come from old customers. Hence, merchants can focus their detection effort on purchases made by *new customers*.

In summary, we want to detect transactions originating from *new* customers that *deviate* from *the norm*, as this is the characteristic of organized fraud. This approach is meant to supplement, and not replace, existing fraud detection, especially to proactively detect emerging forms of fraud. To accomplish that, we will use diversity index to model the 'normal' transactions, and use Mean Absolute Percentage Error (MAPE) to quantify the deviation.

Other types of transaction fraud, such as the so-called friendly fraud [18], are carried out individually, therefore they cannot be reliably detected as they are indistinguishable from legitimate transactions. Furthermore, these types of fraud are small in number as they usually involve the cardholder himself or someone he personally knows, so we do not consider this type of fraud in our research.

IV. Proposed Proactive Fraud Detection Model

In this section, we will explain the concept of device attributes and show how we can use them in conjunction with the Shannon Diversity Index and Mean Absolute Percentage Error in detecting the type of fraud outlined in the previous section.

A. *Device Attribute*

Merchants can only successfully detect fraudsters if they exhibit characteristics that are different than legitimate buyers, particularly the use of the same (or obfuscated) devices under a different identity. Therefore, merchants will collect various device attributes to help support this detection effort, which can be done using Javascript [19], for example. Merchants want to collect as many independent device attributes as possible since it is unlikely that fraudsters will be successful in obfuscating all of these attributes. Some of the common device attributes collected are: IP address (and its derivation, such as ISPs and geo location), browser information (language, user agent, etc.), and device information (OS, time zone, etc.).

B. *Diversity Index*

Fraudsters need to monetize their stolen cards quickly, which naturally leads to the velocity based detection rule [16]. Unfortunately, fraudsters understand this and are able to defeat it by obfuscating the attribute that is used to correlate these fraudulent transactions together. Fraudsters' obfuscation attempts create a conundrum for merchants. If the chosen correlation attribute is too specific (i.e., email, IP address), then fraudsters can easily circumvent it by simply changing that value. With the advent of virtualization and cloud computing, even a previously immutable characteristic of a device, such as browser fingerprint or hardware profiling, can be circumvented by generating new instances of virtual machines. On the other hand, if the fraud detection correlation is too generic (i.e., OS version, browser language), then the correlation will be too weak, resulting in too many false positives.

We propose a unique way of combining the strength of both generic and specific correlation by using what we call *invariant diversity*. It works based on the concept that *the more generic an attribute is, the more diversity will be seen on transactions associated with that attribute*. For example, as an OS version is a rather generic attribute, we should expect to see significant variations for transactions with the same OS version. In other words, the transactions with the same OS version will have numerous different IP addresses, ISPs, emails, items bought, billing regions, etc. If, however, recent transactions from buyers with the same OS version do *not* exhibit this diversity, then this is a very strong indication of artificial manipulation and fraudulent activity.

Diversity can be quantitatively measured using a *diversity index*, which is normally used by ecologists to calculate how evenly the *species* are distributed in a *community*. The most popular index is the Shannon Diversity Index [20] shown in Eq. 1, where p_i represents the proportion of a species to community x and R_x represents the distinct number of species in that community.

$$H'(x) = -\sum_{i=1}^{R_x} p_i \ln p_i \tag{1}$$

Our application of this formula in the fraud detection model is as follows. First, for every transaction attribute pair (x,y), we will use the number of transactions that have a distinct value of y to calculate the diversity index for every specific value x_n. That is, we treat all transactions with the same attribute value x_n as a 'community', and each distinct y_m within that community as a 'species'. For example, in attribute pair (*os_version*, *ip_country*), x could be {Android 5.3, Windows 7, etc.} and y could be {US, GB, MX, etc.}. And for this attribute pair, the value of diversity index H'(Android 5.3), will measure the diversity of countries (*ip_country*) on transactions with *os_version* = Android 5.3. The diversity index should vary for different communities, but it is expected that the larger the community size (R), the more diverse it will be. That is, $H'(x_n) > H'(x_{n'})$ if $R_{xn} > R_{xn'}$.

However, to be able to detect deviation associated with fraudulent transactions, we must first be able to generate a regression function for the diversity index. In order to do this, we want to find a stable invariant-diversity attribute pair (x,y) such that:

$$H'(n) \approx H'(n') \;\forall R_n = R_{n'} \tag{2}$$

$$H'(n) \approx a + b \ln R_n \tag{3}$$

Eq. 2 allows us to generalize the diversity index value without the need to obtain it for every single value of n; that is, if $R_{Android5.3}$ is 3, and H'(Android 5.3) = 0.6, then if $R_{Windows7}$ is 3, H'(Windows 7) ≈ 0.6. Furthermore, Eq. 3 means we can create a regression function of H' without obtaining a data point for every single value of R_n. The value for constants a and b themselves can be resolved using best fit equations in Eq. 4 and Eq. 5, where x is R_n and y is $H'(n)$:

$$b = \frac{n \sum_{i=1}^{n}(y_i \ln x_i) - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} \ln x_i}{n \sum_{i=1}^{n}(\ln x_i)^2 - (\sum_{i=1}^{n} \ln x_i)^2} \quad (4)$$

$$a = \frac{\sum_{i=1}^{n} y_i - b \sum_{i=1}^{n}(\ln x_i)}{n} \quad (5)$$

So, our detection model for attribute pair (x,y) is complete once we obtain the corresponding a and b for the pair. The model will form a regression function of expected diversity index for a given number of transactions that match the attribute pair. Subsequent transactions will be tested for conformance against this model, and extreme deviation can be marked as potential fraud.

C. Mean Absolute Percentage Error

The next step in building our model is to select the appropriate attribute pairs to be used in detecting fraud. In order to make it more difficult and time consuming for the fraudsters to successfully obfuscate their device's attributes, we should try to collect as many of these attributes as possible. This will undoubtedly increase the time it takes to create the models as more attribute pairs need to be considered. For example, if we can obtain N different attributes from a customer's transaction/device, there will be overall $N(N\text{-}1)$ different attribute pairs to consider, and hence, $N(N\text{-}1)$ of diversity indices to compute.

Model generation only needs to be done periodically, which makes it less sensitive to increased running time. Performing the detection, however, requires measuring the deviation of a current transaction's attribute pair diversity index against the models. As the number of attribute pairs grows, it becomes too time-consuming to calculate the diversity index deviation for every single attribute pair combination of the transaction under review. Furthermore, some attribute pairs lack any diversity and therefore should not be included in our model.

In order to select the best attribute pairs to be included in our invariant diversity model, we calculate the Mean Absolute Percentage Error (MAPE) [21] of an attribute pair's regression line against the transaction's actual diversity index for the pair. MAPE equation is shown in Eq. 6, where F_i is value from regression function, and A_i is the actual diversity index for this attribute pair. We will select the attribute pair that has the smallest MAPE as it indicates a minimal deviation between the function and actual value.

$$M = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right| \quad (6)$$

MAPE is also used when analyzing a transaction for possible fraud. The same process is performed to obtain the diversity index for this transaction's attribute pair, and the deviation between this pair's diversity index is compared against the value from a regression function relative to the pair's MAPE. Material deviation between the values indicates a lack of diversity, which could indicate tampering with transaction data and is a possible indication of fraud.

V. Implementation

We implement our detection model on real transaction data, courtesy of our online merchant partner, MaximusCards (www.maximuscards.com). We start by selecting the list of attributes to be considered in the attribute pairs, and deciding on various threshold parameters for the model itself.

A. Device Attribute Selection

For each transaction, there are 32 device attributes that were collected. We remove an attribute from consideration if it is:

- Too rare: attribute is missing in more than 50% of transactions.
- Too unique: Each unique value for that attribute appears in less than two transactions on average.
- Too common: Each unique value appears in > 4% of all transactions.

The final attributes used are shown in Table 1.

TABLE I. List of Device Attributes used in detection

Browser	*Browser_language*
Browser_string_hash	*Browser_version*
DNS_IP_Geo	*DNS_IP_ISP*
Headers_order_string_hash	*JS_browser*
JS_browser_string_hash	*JS_OS*
OS_Version	*Screen_Res*
Time_Zone	*True_IP_Geo*
True_IP_ISP	*UA_Platform*
UA_Browser	

Most of these attributes are self-explanatory, and in any case the model does not require any understanding of the attribute as the formula only looks into the number of transactions per distinct values (i.e., R_n), not the value itself.

B. Dataset size

Although lack of diversity created by fraudulent transactions will become more apparent over a larger number of transactions, nevertheless the fraud detection objective is to detect this in as few transactions as possible to reduce fraudulent loss by an online merchant. Hence, we need to use a dataset size where the number of transactions per unique attribute value is low for at least some of the data points. If the dataset used for creating the model is too large, then resulting regression function might be too inaccurate for smaller R_n values, which means that any deviation (i.e., fraud) can only be detected once the fraudsters conducted significant number of transactions, which is undesirable. So although this seems counterintuitive, using a dataset that is too large is actually detrimental to the model's performance.

Therefore, for transaction pair (x,y), we need to choose a training period that is large enough to contain diverse values of x, but yet short enough so the median number of transactions per unique value (R_x) is below 15. That is, half the dataset are small values so the merchant can detect deviation more precisely without suffering too many fraudulent transactions. Based on our experiments on the 129,116 real transactions provided by our online merchant partner, we choose 7 days as the optimal training period for our model. Later, we are also using this same period during the detection phase when we run the model against every transactions and calculate the diversity index for them.

C. *Attribute Pair Selection*

Next, we create a cross product of each possible combination of attributes, generate the regression function and choose several pairs which have the smallest MAPE for their regression function. Initially, any attribute pair with low diversity will be eliminated. Here, we set the threshold for low diversity as having diversity index of 0 in at least 50% of the data points.

It should be noted that any fraud in the pair's dataset will increase the MAPE of that pair. This is ironic because it means that a pair with larger MAPE might actually be a better model. To fix this problem, once we find the regression function, we will eliminate 8% of data points with the largest Absolute Percentage Error and recalculate the regression again. The number 8% was chosen based on our merchant partner's claim about the average percentage of fraudulent card transactions in the industry.

Finally, we have to choose the number of attribute pairs to be included during the detection phase. More pairs will lead to more sensitive detection, but it will lead to more false positives and longer processing time as well. In the end, we decided to use five attribute pairs with the smallest amount of MAPE. Also, in order to make the attribute pair more diverse, once an invariant attribute is selected, subsequent pairs will not use this same invariant again.

D. *Deviation Detection*

We choose the threshold of two times of MAPE of the pair before we flag the transaction. This will reduce the number of false positives but more fraudulent transactions will happen before it is detected, as illustrated in Figure 1.

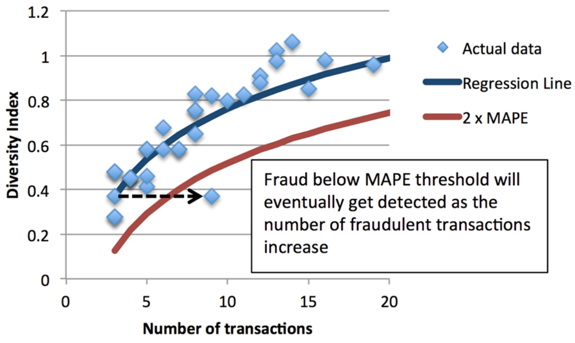


Fig. 1. Illustration of how fraud can eventually be detected due to its deviation become more apparent as more fraudulent transactions are generated by the same fraudster

VI. Real Deployment Results

A. *General Result*

To evaluate the performance of our model, we performed model generation and transaction detection each day for 30 days. The model is generated by analyzing the diversity index of all possible attribute pairs (all candidate attributes are listed in Table 1) of the previous 7 days of transactions, generating the corresponding regression functions and their MAPE, selecting the best five attribute pairs, and then running these models against transactions in the next day. The number of runs is somewhat limited as each potentially fraudulent transaction that was previously undetected must be analyzed manually by our online merchant. The result of these 30 daily runs are summarized in figure 2.

We flagged 1,002 transactions out of 9,547 transactions processed on our partner online merchant in that 30 days period. Of the 1,002, there are 673 transactions that were cancelled by the payment processor due to various, but mostly fraud-related reasons, and there are 216 that were already flagged by the merchant's existing fraud detection methods. Of the 113 new cases of potential fraud, 37 of them are transactions that coincidentally shared an attribute with previous fraudulent transactions. Hence, although it looks like a false positive, the merchant does want to be notified of these transactions as they could potentially come from the same fraudsters. Of the 61 transactions that are truly false positive (i.e., the merchant does not consider them worthy of further review), 36 of them are due to a unique circumstances of certain region in Middle East where there is limited diversity. This is a very specific case, and the merchant can simply create a rule to handle this extraordinary fact. Finally, there are 15 transactions that are truly suspicious and definitely warrant manual review by the merchant.

As our objective is to complement available detection tools and not to replace them, we intentionally do not analyze the number of false negatives in our result. Our system is designed to proactively detect emerging fraud trends without supervision, and actually will not be able to detect trivial frauds such as the use of proxy to hide IP address, for example.

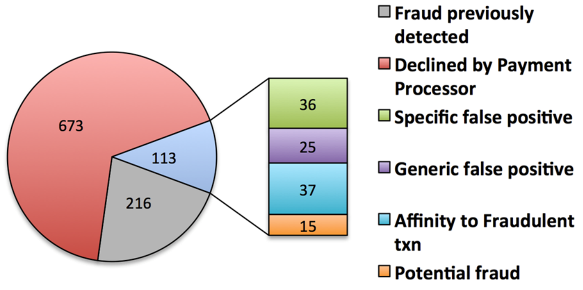
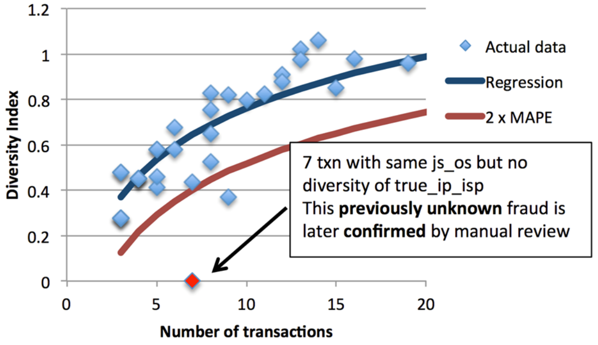


Fig. 2. Result summary

Overall, the result is very promising as only 6% is considered by the merchant to be truly false positive. And if we eliminate the one specific regional case, then the false positive is only 2.5%. Our model is created without any supervised input or training, which makes this result even more impressive. Furthermore, there are cases of previously unknown and potentially fraudulent transactions that the merchant is very keen on analyzing further.

B. *Newly discovered fraud*

We want to analyze one newly discovered fraud that is detected by our model and confirmed by the merchant. In this case, the transaction seemed benign as it was coming from a new user with a new device that was not related by IP address, name, email or credit card number to any previous transactions. Using our detection model, however, this transaction shows significant deviation from the expected diversity index value expected of community with the same number of transactions (as shown in figure 3), and hence it was flagged as a possible fraud.



7 txn with same js_os but no diversity of true_ip_isp
This **previously unknown** fraud is later **confirmed** by manual review

Fig. 3. Newly discovered fraud by invariant diversity of (js_os, true_ip_isp)

The detection process for this transaction *x* is as follows:

- Analysis of transactions from previous 7 days prior to *x* produce five invariants - diversity attribute pair model. In particular, the pair (js_os, true_ip_isp) has a regression function of $0.011 + 0.326\ln(R_n)$, and MAPE of 0.122.

- The *js_os* of *x* is 'Android 4.3'. In this case, $R_{\text{Android 4.3}}$ happens to be 7 (i.e., there are 7 transactions in the last 7 days that has *js_os* of 'Android 4.3'). Therefore, it is expected that H'(Android 4.3) with diversity attribute *true_ip_isp* is $0.011 + 0.326\ln(7) \approx 0.646$. However, all 7 transactions have the same *true_ip_isp*, hence $H' = 0$. Since MAPE is 0.122, any H' below $0.646 - 2*0.122 = 0.402$ is considered abnormal and will be flagged.

This fraudster reset his Android device every time he made a purchase with a new stolen credit card, but since his OS and his ISP stayed the same, our model considers this to be abnormal as it expects a *diverse* ISPs and hence flagged the transaction. After we showed our finding, the merchant conducted manual review of these transactions and positively confirmed that they are indeed fraudulent transactions that were previously undetected. Hence, although this type of obfuscation is very difficult to detect using a common rule-based approach, our invariant diversity method can detect it without any supervision.

C. Future Work

This is our first attempt to use invariant diversity approach to detect abnormalities in transactional data, and considering the promising results, there are several future research ideas that can be pursued. First, we want to analyze if any of the parameters that we choose manually, such as the length of training dataset, the selection of attribute to be discarded, etc., can be automated. Second, we want to improve the selection of attribute pair and the calculation for the deviation to achieve a better abnormalities detection. Finally, we want to conduct research to see if this invariant diversity approach can be used to detect abnormalities in other web-based user application, such as review websites.

VII. Conclusion

In this paper, we have demonstrated how an online merchant can use "invariant diversity" to proactively detect fraud with a small number of training datasets in an unsupervised environment. Our method has been tested against real transactional data and yielded exceptional results with the ability to detect even previously undetected fraudulent transactions.

References

[1] TSYS, "2014 Consumer Payments Study", October 2014.

[2] K. Neisen, "EMV and payment card fraud: the impact of EMV on fraud trends," The Copper River Group, August 2015.

[3] J. Conroy, "Card-not-present fraud in a post-EMV environment: combating the fraud spike," Aite Group, June 2014.

[4] MasterCard, "MATCH", https://developer.mastercard.com/documentation/match/ , 2016.

[5] http://www.zdnet.com/article/visa-pushes-for-two-factor-authenticated-transactions/

[6] http://www.computerweekly.com/feature/Two-factor-authentication-Next-step-to-secure-online-transactions

[7] C. Whitrow, D. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," Data Mining and Knowledge Discovery, vol. 18, no. 1, pp. 30–55, Jul. 2008.

[8] S. Ghosh and D.N. Reilly, "Credit card fraud detection with a neural network," Proceedings of 27th Annual Hawai International Conference on System Sciences, 1994.

[9] R. Brause, T. Langsdorf, M. Hepp, "Neural data mining for credit card fraud detection," Proceedings of 11th IEEE International Conference in Tools with Artificial Intelligence, 1999.

[10] A. Brabazon, J. Cahill, P. Keenan, D. Walsh, "Identifying online credit card fraudusing artificial immune system," IEEE Congress on Evulotionary Computation, 2010.

[11] E. Duman, M.H. Ozcelik, "Detecting credit card fraud by genetic algorithm and scatter search," Expert System with Application Vol. 38, No. 10, 2011.

[12] E. Lee, J. Woo, H. Kim, A. Mohaizen, H.K. Kim, "You are a game bot!: uncovering game bots in MMORPGs via self-similarity in the wild," Network and Distributed System Ssecurity Symposium 2016, California, 2016.

[13] R.J Bolton, D.J. Hand, "Unsupervised profiling methods for fraud detection," Proc. Credit Scoring and Credit Control VII, 2001.

[14] A. Srivastava, A. Kundu, S. Sural, A. Majumdar, "Credit card fraud detection using Hidden Markov Model," IEEE Transactions on Dependable and Secure Computing Vol. 5, No. 1, 2008.

[15] Y. Dai, J. Yan, X. Tang, H. Zao, and M. Guo, "Online card fraud detection: a hybrid framework with big data technologies," IEEE Trustcom/BigdataSE/ISPA, Tianjin, China, 2016.

[16] K. R. Seeja, Masoumeh Zareapoor, "FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining", *The Scientific World Journal*, September 2014.

[17] ThreatMetrix, "Device Fingerprinting", https://www.threatmetrix.com/wp-content/uploads/2010/03/ThreatMetrix_Datasheet.pdf, 2010.

[18] Experian, "Friendly Fraud: When the Thief is a Friend, Family Member or Acquaintance", https://www.protectmyid.com/images/education_center/pdf/050TypesofFraud/4_types%20of%20fraud_friendly.pdf, 2010.

[19] K. Mowery, D. Bogenreif, S. Yilek, H. Shacham, H. Wang, "Fingerprinting information in JavaScript implementations", *Proceedings of W2SP 2011. IEEE Computer Society*, May 2011.

[20] Anne Magurran, "Measuring Biological Diversity", Wiley-Blackwell; 1st edition, December 19, 2003.

[21] A. De Myttenaere, B. Golden, B. Le Grand, F. Rossi, "Using the Mean Absolute Percentage Error for Regression Models", Computational Intelligence and Machine Learning (ESANN), Apr 20.