W3Detector: Detecting Fraudulent Online Sellers Based on Temporal and Spacial Information

Shengbo Tong*, Yuyang Xie*, Shenghua Liu†‡, Wenjian Yu*, Jixuan Cai§
*Dept. Computer Science & Tech., BNRist, Tsinghua University, Beijing, China.
†Insititute of Computing Technology, Chinese Academy of Sciences, Beijing, China.
‡University of Chinese Academy of Sciences, Beijing, China.
§ Tencent Inc., ShenZhen, China.
{tsb23, xyy18}@mails.tsinghua.edu.cn, liushenghua@ict.ac.cn, yu-wj@tsinghua.edu.cn, codyjxcai@tencent.com

2023 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-4534-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICMLA58977.2023.00280

***Abstract*—With the rapid development of electronic payment, risks in large-scale financial transaction networks are persistently arising. One important problem is how to use the transaction data to detect sellers who do fraudulent trades on the online platform. This paper analyzes three attributes of the transaction flow: money (what), time (when), region (where), and summarizes corresponding characteristics among fraudulent sellers. Based on this, we propose an unsupervised anomaly detection algorithm: W3Detector. It processes the raw data with a new sequence discretization method and uses statistical tools to convert data indicators into information value. With the minimum description length principle, we finally filter out the suspicious sellers. The algorithm is not affected by factors such as unit or range of the attributes, and is highly flexible and scalable. After running on three real-world transaction datasets collected from WeChat Pay, W3Detector has show significantly better performance than the baseline methods of anomaly detection, with weighted accuraccy (WACC) increased by 1.13X and F-score increased by 0.105 on average, and the recall value over 60%.**

***Index Terms*—anomaly detection, fraudulent trade, unsupervised algorithm, e-commerce business**

I. Introduction

With the popularity of e-commerce business, launching a new seller account has never been simpler. However, along with the convenience comes potential risks, as fraudsters can easily pose the legitimate seller platform to deceive consumers and make illegitimate profits, which causes significant financial losses, harm to reputation, and legal penalties.

According to previous research findings, the fraudulent sellers and risky transactions can be detected by analyzing the abnormal patterns. However, many anomaly detection methods in online transactions focus on the transfers of money between homogeneous accounts [1], [2], but the scenario under which the money flow between sellers and consumers is unidirectional and only seller accounts are what we concern. Additionally, the platform, acting merely as a transaction channel, cannot obtain other relevant information (such as chat history) before and after the transaction. All data we can use in this problem is the transaction records between sellers and consumers, in which the flow of money cannot be traced.

Supervised learning methods are also not suitable for this problem. In actual business situation, it is hard to collect enough labels, and only a small part of deceived consumers will report the scam experiences which may drown in a plethora of other bogus complaint records. Manually labeling these seller accounts is also very time-consuming and costly [3]. This leads to the lack of labels and mislabeling, which will easily cause overfitting. Moreover, supervised models have to be retrained frequently due to the rapid shift of scam tricks and new labels [4].

Upon this background, we propose W3Detector, an unsupervised anomaly detection algorithm that can detect suspicious sellers accurately and quickly by focusing on the money (what), time (when), region (where) of transaction records. The main contributions of this work are:

- Four key traits of the online fraudulent sellers are discovered, for each of which a suitable data indicator is proposed. Specifically, a sequence discretization approach considering the context correlation of money values is proposed for data preprocessing.
- The minimum description length principle is leveraged to transform the data indicators into information values, based on which we propose an algorithm W3Detector to detect e-commerce fraudulent sellers. W3Detector is unsupervised, and suitable for being deployed on distributed computing platforms.
- With real-world online shopping transaction datasets, we have validated the effectiveness of W3Detector. Compared with the existing unsupervised anomaly detection methods, W3Detector makes WACC increased by 1.13X and a 0.105 larger F-score, on average.

II. Related Work

In this section, we will review two categories of related works on anomaly detection.

A. Time-series Methods

There have been many studies that focus on data mining over time series. RSC [5] uses a generative model to find out the inter-arrival time characteristics, in which there are significant differences in the distribution patterns of temporal activities between humans and bots. MIDAS [6] focuses on detecting anomalies within the data stream, by identifying groups of incoming edges within a short period of time. This method detects microclusters online by employing the Chi-square test, which guarantees its theoretical precision, and only

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00280

consumes constant memory and time. Matrix Profile [7] aims to detect anomalous segments in the data sequence. By sliding a window over the data, the distances between subsequences are calculated in order to identify motifs and discords. The distance matrix can also be used in other data mining tasks.

Attempting to apply these works to the problem addressed in this paper, we found the following issues. Accounts of fraudulent sellers are also operated by humans, and their behavioral patterns do not exhibit the same level of randomness as bots. This makes it difficult to distinguish them using the RSC [5] model. Moreover, money sequences differ from variables such as water pressure and electric current, whose fluctuations may not correspond to anomalous events, rendering the MIDAS [6] and Matrix Profile [7] ineffective.

B. Graph and Tensor Models

Many methods use matrices to formulate the similar kind of problems. By treating subjects and timestamps as the two dimensions of a matrix, anomaly detection can be formulated as a community detection problem. EigenPulse [8] proposes a dense subgraph detection approach for row-augmented matrices, which employs a single-pass randomized singular value decomposition algorithm [9] to identify suspicious groups among the top singular vectors with large absolute values [10]. Another approach [3] is to construct a bipartite graph between subjects and their attributes, and then compute anomaly scores based on different rules. Weights are updated through a linear propagation algorithm [11], and the sum of weights shared by each pair of nodes are mapped to a new weighted graph, in which community detection algorithms [12] can be used to identify dense subgraphs.

Another type of methods extend the problem of detecting dense sub-blocks to high dimensions, using tensors instead of matrices to store multi-dimensional data records. M-Zoom [13], M-Biz [14] and D-Cube [15] provide a definition of density for tensors and selects the top dense sub-tensors by a greedy algorithm with high quality ensured. AugSplicing [16] improves the performance by identifying a necessary and sufficient condition for expanding dense blocks on which the maintaining of a maximum heap is based, thus significantly accelerating computation time and enabling the handling of streaming data that grows with time.

However, there are all certain limitations of these methods. One is that they all tend to rely on the characteristic of density, but not all anomalies actually manifest as denseness in some dimensions, and may instead be some kind of sparseness or outlier. Another constraint is that they handle different features in a uniform manner and are unable to make corresponding adaptations based on different patterns of each attribute.

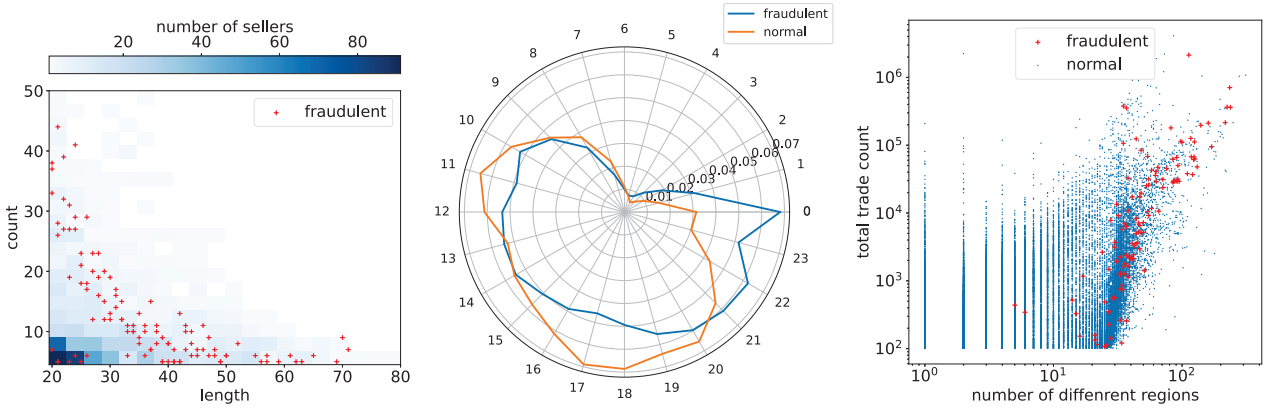
III. Data Observation

We analyze the characteristics of money, time and region in transactions, and discover four key traits of fraudulent sellers.

1) Money pattern: The features of fraudulent sellers in transaction value can be divided into two categories: local and global.

Local feature refers to conducting similar transactions with different consumers. This may be a scamming technique used by fraudulent sellers to lure victims into a trap. It should be pointed out that this feature is only effective after the sequence reaches a certain length. Otherwise, maybe it is just because they are selling a specific type of product. After grouping and counting different money sequences for each seller, we calculate the maximum occurrences of sequences that have the same length, and plot a heatmap in Fig. 1(a). The positions where fraudulent sellers appear are marked, and it can be observed that they have long sequences with a large number of repeated transactions.

Global characteristic refers to sellers conducting extremely rare or special transactions among the entire population. There are two main reasons: one is that the value has a special meaning or is a discount price, which is a way to inveigle the victims; the other is that the value is just below a certain threshold, such as the maximum limit for payment channels like electronic transfers or bank cards, leading fraudulent sellers to offer prices slightly below the line. These types of



(a) money repetition (b) time abnormality (c) region dispersion

Fig. 1: Anomalous features of fraudulent sellers, in comparison with normal sellers.

money sequences are not usually found among normal sellers. For example, "199, 299, 399" is obviously not likely to be a normal transaction record.

2) Time abnormality: Fraudulent sellers have different preferences than normal sellers in terms of transaction time. They are more likely to do fraudulent trades during non-working hours, when victims are relatively more available to interact with them. To verify this feature, we plot a radar chart in Fig. 1(b). The outer circle in the chart represents 24 hours, and the distribution of total transaction number in each seller group is normalized. Normal sellers have the highest transaction proportion in the evening (5 P.M.-8 P.M.), while fraudulent sellers have the highest transaction proportion in the late night (10 P.M.-1 A.M.).

3) Region dispersion: Fraudulent sellers tend to have more dispersed transaction regions. This can be explained by the fact that fraudulent transactions have lower costs, so distance is not an important factor to consider. In contrast, for normal sellers, do trades with local consumers can increase consumer loyalty, while fraudulent sellers are more likely to face complaints and investigations. To verify this feature, we draw a scatter plot in Fig. 1(c) for the number of different regions and the total trade count of sellers. Fraudulent sellers tend to do trades in more than 10 regions, some even up to 100, with a relatively large trade volume.

IV. Proposed Method

In this section, we first give the problem definition and algorithm framework. Then, we devise different data indicators on money, time and region. Finally, we employ the minimum description length principle to combine all the indicators, and propose the W3Detector algorithm.

A. Problem Definition and Algorithm Framework

Since we focus on the money, time and region of transaction flow, we define our problem as follows:

Problem 1 (fraudulent online sellers detection): Given a stream of transaction records $\{r_1, r_2, \cdots\}$, where each item $r = (u, v, m, t, l)$ represents a transaction, consisting of the seller u, consumer v, amount of money m, timestamp t and location l. Money and timestamp are measured in cents and seconds respectively, while location represents the name of regions. **Our goal is to find group $\mathcal{B}$ of suspicious sellers that meet the above four traits.**

For each of the traits, we can devise different indicators of the raw data. In order to combine them together, we introduce the Minimum Description Length (MDL) [17] principle. It is a criterion primarily used for model comparison and selection. Formally, the principle states that given a family of models $\mathcal{M}$, the best model $M \in \mathcal{M}$ for data D is given by:

$$\arg\min_{M \in \mathcal{M}} L(M) + L(D|M) \ . \tag{1}$$

Two typical implementations for MDL are Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC):

$$\mathrm{AIC} = 2k - 2\log p(y|\hat{\theta}_{MLE}) \tag{2}$$

$$\mathrm{BIC} = k\log n - 2\log p(y|\hat{\theta}_{MLE}) \tag{3}$$

where y, k, n, $\hat{\theta}_{MLE}$ and $p(\cdot)$ are the sample value, the number of parameters, the size of sample, the maximum likelihood estimation of the parameters and the probability density function of the model, respectively. The first term refers to the complexity of the model, while the second term refers to how well the model fits the data.

It is worth noting that given a specific model, the second term also reflects the probability of the data occurrence, and thus can be used to describe the anomaly degree of a data point. For all the four indicators, we first fit each of its distributions respectively and choose the best model by applying the MDL principle. Afterwards, since AIC and BIC share the same second term, we can use $-2\log p(y|\hat{\theta}_{MLE})$ to unify all the indicators into description length. Sellers with the largest total description length are considered to be suspicious. The whole algorithm framework is shown in Fig. 2.

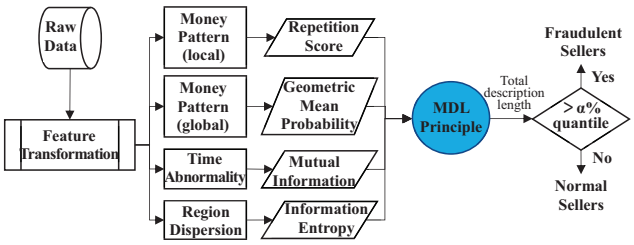


Fig. 2: W3Detector algorithm framework.

B. Feature Transformation

Before modeling the data indicators, it is necessary to perform feature transformation by discretizing continuous attribute values. For time attribute, it becomes an enumerated type after being divided into 24 hours; for region attribute, it belongs to an enumerated type itself. However, unlike time and region, money attribute does not have a natural partition, so we propose a new discretization approach.

We first use the concept of word embedding: the value of money is treated as a token and the transaction sequences between each seller and consumer become corpus composed of these tokens. Using this as the input data, the Word2Vec [18] model is used to train the embeddings of each value. After that, we continue to change vectors into letters by using k-means, with the number of clusters $k = 10$. By mapping each cluster to a letter, different values of money will be classified into ten groups from a to j.

The aim of this method is to break the tradition of separating bins evenly on width or frequency [19], because we think money is different from other common values. Money is always associated with goods or services, and its values

contain certain information. For example, 999 is likely to be connect with promotional discounts but 1000 may be a fixed fee such as a ticket, which have obvious differences in scenarios. In this sense, 899 is more similar to 999 than 1000. This approach is able to achieve this goal because the Word2Vec model can discover the context correlation of transactions, and the similarity of embeddings obtained from Word2Vec can be measured by cosine distance.

The discretization method also has the following three benefits:

- It is robust because the value of money can fluctuate within a certain threshold.
- It largely reduces the magnitude of different money values, alleviating the redundancy of representations.
- It changes the type of money from float to char and cuts down the storage cost.

C. Data Indicators

In this part, we will calculate four data indicators to reflect characteristics of a seller on the four traits.

1) Money pattern: The local and global money patterns can be handled separately.

Local: The entire transaction dataset can be processed in the following steps: 1. sort the dataset in time order; 2. merge each transaction between a seller and a consumer into a sequence; 3. remove all sequences whose length is less than 3; 4. group and count all sequences for every seller, recording the sequence length (l) and occurrence frequency (f); 5. from all sequences of the same length, select one with the largest frequency. Then we can define the repetition score (S):

$$S = l \cdot \log f \tag{4}$$

The formula has two characteristics: the length of a sequence is more important than its frequency; sequences that only appear once will have no contribution. For each seller, we select the maximum S among all sequences.

Global: To find special transactions, we change the goal into filter out the sequences that are not likely to appear. After the discretization step, all transactions become sequences composed of finite states. Considering computational resources and practical scenarios, we use a second order Markov Chain model to form the problem. The initial distribution and stochastic matrix are calculated through the maximum likelihood estimation. Each sequence (s) now corresponds to a probability of occurrence ($p(s)$), but generally longer sequences have lower probabilities. In order to compare the anomaly more fairly, the geometric mean probability ($\overline{p}$) should be calculated based on the length (l):

$$\overline{p} = -\log\left(p(s)^{\frac{1}{l-1}}\right) = -\frac{\log p(s)}{l-1} \tag{5}$$

Logarithm is used to avoid underflow caused by plenty of multiplications when calculating $p(s)$. Sequences with large $\overline{p}$ are more special, so we select its largest value of each seller.

2) Time abnormality: After dividing timestamps into 24 hours, we can obtain the distribution of all transactions in one day. We separately compute the transaction distribution of each seller, and calculate the *mutual information* between it and the total distribution. Since normal sellers have almost no transactions during abnormal times, fraudulent sellers have a relatively higher correlation with the overall distribution, and therefore gets a larger value.

3) Region dispersion: After counting the number of transactions in different regions, we can obtain the probability distribution of each seller. To measure its dispersion, we calculate the *information entropy* of this distribution after normalization, where a larger value shows that trades occur in more regions separately.

D. W3Detector Algorithm

The pseudo code of W3Detector algorithm is shown in Algorithm 1. We notice that for all the four indicators, only large values reflect higher suspicion, but by using the formula $-2\log p(y|\hat{\theta}_{MLE})$, small values will also get a long description length when the distribution function has a maximum value, so we make a little adaption (see line 11-15), where small values will get the least description length in this case. The models we choose have no more than one maximum value, so only unimodal distributions (see line 9) will meet this condition.

Algorithm 1 W3Detector: Fraudulent online sellers detection

Input: Transaction records $\mathcal{R} = \{r_1, r_2, \cdots\}$
Output: Suspicious sellers group $\mathcal{B}$

1: Use the money sequences to train a Word2Vec model
2: Run k-means algorithm on the word embeddings
3: Discretize records $\mathcal{R}$ on three attributes
4: Compute four data indicators on money, time and region
5: $\{\mathcal{L}_j\}_{j=1}^{n} \leftarrow \mathbf{0}$ ▷ n is the number of sellers
6: **for** $i = 1 \to 4$ **do** ▷ iterate indicators to get total length
7: Fit a family of alternative models $\mathcal{M}$ for $\mathcal{A}^{(i)}$
8: Choose the best model $M \in \mathcal{M}$ according to MDL
9: $unimodal \leftarrow \max p(\cdot|\hat{\theta}_{MLE}) < \infty$
10: **for** $j = 1 \to n$ **do**
11: **if** $unimodal$ **and** $\mathcal{A}_j^{(i)} < \arg\max p(\cdot|\hat{\theta}_{MLE})$ **then**
12: $\mathcal{L}_j \leftarrow \mathcal{L}_j - 2\log\left(\max p(\cdot|\hat{\theta}_{MLE})\right)$
13: **else**
14: $\mathcal{L}_j \leftarrow \mathcal{L}_j - 2\log p(\mathcal{A}_j^{(i)}|\hat{\theta}_{MLE})$
15: **end if**
16: **end for**
17: **end for**
18: Select top $\alpha\%$ sellers $\mathcal{B}$ with the largest $\mathcal{L}$
▷ α is a hyper parameter depends on the actual situation
19: **return** $\mathcal{B}$

Only statistical computations and two common tools (Word2Vec, k-means) are involved, so it is easy to deploy W3Detector on distributed platforms to process big data.

V. EXPERIMENTS

A. Experiment Setup

Datasets: Three real-world transaction subsets collected from WeChat Pay are used in this work: Media, Business and Service. Low active sellers and extreme huge sellers are excluded to avoid interference to data distributions.

- Media: Payments to online video platforms, live streaming and online games.
- Business: Payments to online shopping platforms.
- Service: Payments to food, recreation and other local services.

The statistical information of the three datasets is listed in Table I. The number of fraudsters corresponds to the sellers who are complained by consumers and then verified as fraudulent sellers by the consumer service personnel. This manual verification may be incomplete, but we still use its results the ground-truth in the following experiments.

TABLE I: Statistics of real-world datasets.

Dataset	# of fraudsters	# of sellers	# of transactions
Media	122	41465	132133444
Business	38	13098	40441086
Service	74	23753	100407795

Baselines: M-Zoom [13], M-Biz [14] and D-Cube [15] are tested as the baselines. They output dense blocks and all sellers in them are regarded as the suspicious sellers. We have tried different combinations of hyper parameters (density measure, number of blocks) in the baseline methods and report the best performance of each baseline.

Evaluation Metrics: The problem of detecting fraudulent sellers is similar to classification. So we compute the metrics of true positive (TP), true negative (TN), false positive (FP), false negaive (FN), precision (P), recall (R) of each method. Particularly, we use the weighted accuraccy (WACC) and F-score (F_β) [20] to evaluate the overall performance. These performance metrics are defined as follows.

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \qquad R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{6}$$

$$\mathrm{WACC} = \frac{\mathrm{TP}+\mathrm{FP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}\left(P - \frac{\mathrm{TP}+\mathrm{FN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}\right), \tag{7}$$

$$F_\beta = (1+\beta^2)\cdot\frac{P\cdot R}{\beta^2\cdot P + R}\ . \tag{8}$$

WACC is defined to constrain the scale of intervene, as the second term adds a weight to the accuracy and avoids the model from outputing too much sellers. β is a parameter that controls the evaluation preference. In this problem, recall is more important, so we set $\beta = 5$.

All baselines are carried out on a Linux machine with a 2.5GHz Intel Xeon Platinum 8255C CPU and 40 GB RAM. The programs of baselines are provided by the authors [1, 2], which are implemented in Java. The proposed method W3Detector is implemented in PySpark.

[1]M-Zoom, M-Biz: https://github.com/kijungs/mzoom

[2]D-Cube: https://github.com/kijungs/dcube

B. Experiment Results

We first show the results of feature transformation in money. For Media dataset, the five money values with the largest cosine similarity to 499 and 400 are shown in Table II. From the table, we can see that the values similar to 499 do not include 500, while the values similar to 400 are all multiples of 100. The classification results are shown in Table III. They are obtained with the context-aware Word2Vector and the k-meaning clustering, which looks reasonable.

TABLE II: The largest similarity values to two example money values with the feature transformation approach.

value	498	799	399	497	496
similarity to 499	0.740	0.729	0.724	0.706	0.666
value	300	700	500	800	200
similarity to 400	0.846	0.783	0.774	0.769	0.753

TABLE III: The 10 classes of money values in the Media dataset, obtained via Word2Vector and k-means clustering.

a	b	c	d	e	f	g	h	i	j
499	29800	627500	6800	20000	400	540	500	77220	6990
799	38800	950500	2800	50000	300	660	1000	10010	5980
498	34800	2066000	9800	40000	800	440	5000	19770	7980
...	...	...	...	...	...	...	...	...	...

Next, we show the results of four data indicators defined previously. We take the Service dataset as an example. The distributions of these data indicators are plotted in Fig. 3. For each one we fit it with over 30 probability density functions in scipy module including `beta`, `gamma`, `exponential`, `pareto`, `laplace`, `inverse gaussian`, etc. Considering the metrics of both AIC and BIC we finally select the best fitting models for them, which are also plotted in Fig. 3. From the figure we can see that the selected models are able to fit the distributions to a satisfactory level.

Finally, the results of W3Detector and the three baselines are listed in Table IV. From it we see that W3Detector has much better precision than the baselines while preserving comparable or better recall. With respect to the overall performance metrics, W3Detector makes WACC increased by 1.13X averagely on the three datasets compared to the best of the three baselines, while on average making F_5 0.105 larger than that obtained with the best baselines.

It should be pointed out that the above performance metrics are evaluated based on the verified numbers of fraudulent sellers which may not cover all. For example, in the Media dataset, 822 of all the 2488 suspicious sellers (filtered by W3Detector) have complaint records, but only 88 are labelled. This is the main reason for the low precision of the proposed method. With further work on manual investigation we believe more fraudulent sellers can be found in the datasets thus leading to better performance metrics. Lastly. we also show the running time of W3Detector and three baselines in Table V. W3Detector can finish the task in a rather fast speed.

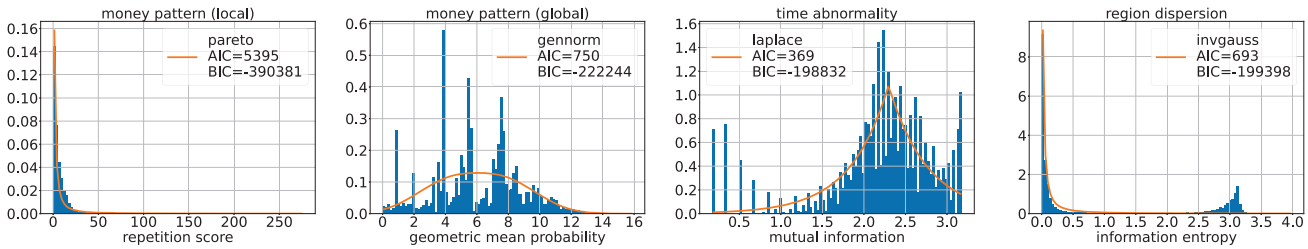


Fig. 3: The distributions and best fitting models for the four data indicators on the Service dataset.

TABLE IV: Results of W3Detector and three baselines for detecting fraudulent sellers.

Dataset	Method	TP	TP+FP	P	R	WACC	F_5
Media	M-Zoom (ari, 15)	77	4798	0.016	0.631	0.00152	0.255
	M-Biz (ari, 15)	89	8171	0.011	0.730	0.00157	0.206
	D-Cube (ari, 15)	81	3686	0.022	0.664	0.00169	0.313
	W3Detector ($\alpha=94$)	88	2488	0.035	0.721	**0.00195**	**0.413**
Business	M-Zoom (geo, 3)	25	4149	0.006	0.658	0.00099	0.127
	M-Biz (geo, 3)	25	4162	0.006	0.658	0.00099	0.127
	D-Cube (geo, 3)	24	2093	0.011	0.632	0.00137	0.205
	W3Detector ($\alpha=94$)	23	810	0.028	0.605	**0.00158**	**0.340**
Service	M-Zoom (ari, 10)	43	1362	0.032	0.581	0.00163	0.348
	M-Biz (geo, 2)	43	1386	0.031	0.581	0.00163	0.345
	D-Cube (ari, 10)	45	1815	0.025	0.608	0.00166	0.319
	W3Detector ($\alpha=96$)	46	951	0.048	0.622	**0.00181**	**0.427**

"(xxx, xx)" indicates the density measure (arithmetic mean or geometric mean) and the number of output blocks.

TABLE V: Running time of W3Detector and three baselines for detecting fraudulent sellers.

Method	Media	Business	Service
M-Zoom	27min1s	4min17s	14min8s
M-Biz	28min25s	4min32s	9min9s
D-Cube	13min9s	1min42s	6min53s
W3Detector	5min34s	3min38s	4min31s

VI. Conclusions

With the increasing scale of electronic payments, fraudulent activities on online financial networks often occur. This paper focuses on the fraudulent online sellers who cheat consumers. They are difficult to be traced, and to be prevented in advance. Based on the observation of a large amount of transaction records, we discover four distinct traits of the fraudulent sellers and handle them with model fitting approaches individually. Then, we propose an unsupervised anomaly detection algorithm named W3Detector based on the minimum description length principle. Experiments on real-world datasets show that W3Detector can identify fraudulent online sellers more effectively than the baseline methods of anomaly detection. The algorithm will be deployed in actual business process, and plays a positive role in technology for the good.

Acknowledgements

This paper is supported by 2022 Tencent Wechat Rhino-Bird Focused Research Program and National Science Foundation of China under Grant No.U1911401, U21B2046.

References

[1] J. Jiang, Y. Hu, X. Li, W. Ouyang, Z. Wang, F. Fu, and B. Cui, "Analyzing online transaction networks with network motifs," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3098–3106.

[2] X. Sun, W. Feng, S. Liu, Y. Xie, S. Bhatia, B. Hooi, W. Wang, and X. Cheng, "Monlad: Money laundering agents detection in transaction streams," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 976–986.

[3] X. Liang, Z. Yang, B. Wang, S. Hu, Z. Yang, D. Yuan, N. Z. Gong, Q. Li, and F. He, "Unveiling fake accounts at the time of registration: An unsupervised approach," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3240–3250.

[4] D. Yuan, Y. Miao, N. Z. Gong, Z. Yang, Q. Li, D. Song, Q. Wang, and X. Liang, "Detecting fake accounts in online social networks at the time of registrations," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 1423–1438.

[5] A. Ferraz Costa, Y. Yamaguchi, A. Juci Machado Traina, C. Traina Jr, and C. Faloutsos, "Rsc: Mining and modeling temporal activity in social media," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 269–278.

[6] S. Bhatia, B. Hooi, M. Yoon, K. Shin, and C. Faloutsos, "Midas: Microcluster-based detector of anomalies in edge streams," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 3242–3249.

[7] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets," in *2016 IEEE 16th international conference on data mining*, 2016, pp. 1317–1322.

[8] J. Zhang, S. Liu, W. Yu, W. Feng, and X. Cheng, "Eigenpulse: Detecting surges in large streaming graphs with row augmentation," in *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part II 23*. Springer, 2019, pp. 501–513.

[9] W. Yu, Y. Gu, and J. Li, "Single-pass pca of large high-dimensional data," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 3350–3356.

[10] B. A. Prakash, M. Seshadri, A. Sridharan, S. Machiraju, and C. Faloutsos, "Eigenspokes: Surprising patterns and scalable community chipping in large graphs," in *2009 IEEE International Conference on Data Mining Workshops*, 2009, pp. 290–295.

[11] B. Wang, J. Jia, L. Zhang, and N. Z. Gong, "Structure-based sybil detection in social networks via local rule-based propagation," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 523–537, 2018.

[12] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[13] K. Shin, B. Hooi, and C. Faloutsos, "M-zoom: Fast dense-block detection in tensors with quality guarantees," in *Machine Learning and Knowledge Discovery in Databases: European Conference*, 2016, pp. 264–280.

[14] K. Shin, B. Hooi, and C. Faloutsos, "Fast, accurate, and flexible algorithms for dense subtensor mining," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 3, pp. 1–30, 2018.
[15] K. Shin, B. Hooi, J. Kim, and C. Faloutsos, "D-cube: Dense-block detection in terabyte-scale tensors," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 681–689.
[16] J. Zhang, S. Liu, W. Hou, S. Bhatia, H. Shen, W. Yu, and X. Cheng, "Augsplicing: Synchronized behavior detection in streaming tensors," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4653–4661.
[17] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.
[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.
[19] H. Guo, B. Chen, R. Tang, W. Zhang, Z. Li, and X. He, "An embedding learning framework for numerical features in ctr prediction," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2910–2918.
[20] V. Rijsbergen, *Information Retrieval*. Butterworths, 1979.