You Are How You Use: Catching Gas Theft Suspects among Diverse Restaurant Users

Xiaodu Yang[1,2,3,4], Xiuwen Yi[3,4,5*], Shun Chen[1,3,4], Sijie Ruan[6,3,4]
Junbo Zhang[3,4,1], Yu Zheng[3,4,1,6], Tianrui Li[1]
[1] Artificial Intelligence Institute, Southwest Jiaotong University, China [2] CentraleSupélec, Université Paris-Saclay, France
[3] JD Intelligent Cities Research, China [4] JD Intelligent Cities Business Unit, JD Digits, China
[5] Department of Computer Science and Technology, Tsinghua University, China
[6] School of Computer Science and Technology, Xidian University, China
{xiaodu.yang,xiuwenyi}@foxmail.com;sjruan@stu.xidian.edu.cn
{bhchenshun,msjunbozhang,msyuzheng}@outlook.com;trli@swjtu.edu.cn

ABSTRACT

Gas theft of restaurants is a major concern in the gas industry, which causes revenue losses for gas companies and endangers the public safety seriously. Traditional methods of gas theft detection highly rely on active human efforts that are extremely ineffective. Thanks to the gas consumption data collected by smart meters, we can devise a data-driven method to tackle this issue. In this paper, we propose a gas-theft detection method *msRank* to discover suspicious restaurant users when only scarce labels are available. Our method contains three main components: 1) *data pre-processing*, which filters reading noises and excludes data-missing or zero-use users; 2) *normal user modeling*, which quantifies the self-stable seasonality of normal users and distinguishes them from unstable ones; and 3) *gas-theft suspect detection*, which discovers gas-theft suspects among unstable users by RankNet-based suspicion scoring on extracted deviation features. By using detected normal users as negative samples to train RankNet, the component of normal user modeling and that of gas-theft suspect detection are seamlessly connected, overcoming the problem of label scarcity. We conduct extensive experiments on three real-world datasets, and the results demonstrate advantages of our approach. We have deployed a system *GasShield* which provides a gas-theft suspect list weekly for a gas group in northern China.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; • **Information systems** → *Information systems applications.*

KEYWORDS

Gas Theft Detection; Utility Fraud Detection; Time Series Anomaly Detection; Non-technical losses; Urban Computing

*Xiuwen Yi is the corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

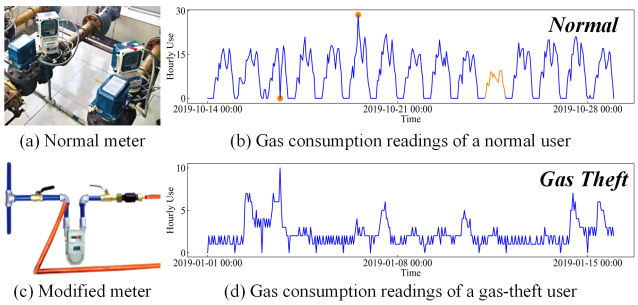
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412751

ACM Reference Format:
Xiaodu Yang, Xiuwen Yi, Shun Chen, Sijie Ruan, Junbo Zhang, Yu Zheng, and Tianrui Li. 2020. You Are How You Use: Catching Gas Theft Suspects among Diverse Restaurant Users. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412751

1 INTRODUCTION

The scale of Chinese catering market has reached 4.2 trillion in 2019 [1], and restaurants account for a significant part of gas users. However, motivated by saving operation costs, the phenomena of gas theft widely exists among restaurant users. To report less charged gas consumption than the actual volume they have used, gas-theft users tend to modify or even destroy gas equipment. Figure 1(a) and (c) show a normal meter and a meter modified by gas-theft users respectively. Given the large number of restaurant users, detecting their gas theft behaviors is of considerable value from the following two aspects. For natural gas suppliers, large-scale gas thefts lead to tremendous revenue losses. For the public, gas theft means can cause gas leakage or even explosions which endanger the public safety seriously, especially when restaurants normally locate in crowded areas of business or residence.

Traditional methods of gas theft detection highly rely on active human efforts, like on-site inspections conducted by operators. Without specific target suspects, these methods can only cover either a small portion of users randomly or all users at low inspection frequency, which are rather ineffective and lagged.



(a) Normal meter (b) Gas consumption readings of a normal user
(c) Modified meter (d) Gas consumption readings of a gas-theft user

Figure 1: Data-driven gas theft detection.

Fortunately, the extensively deployed smart meters report massive gas consumption readings periodically as shown in Figure 1(b) and (d), which brings us an opportunity to devise a data-driven method for detecting gas-theft restaurant users.

However, to detect gas-theft restaurants based on real-world data is non-trivial because of the following two challenges:

- **Limited labels.** Gas theft detection can be categorized as one type of utility fraud detection problems. Existing methods in this domain are either supervised [4, 7, 8, 21] or semi-supervised [13], which heavily rely on the fully labeled data that are synthetic or collected from tens of thousands of on-site inspections [6, 7]. But as in most industrial scenarios, we only have scarcely labeled data. And we cannot collect more labels limited by the high cost of on-site inspections.
- **Complex gas-theft behaviors.** Gas-theft users mostly break the gas equipment, which causes abnormal gas consumption readings occurring at any time in unpredictable forms without consistent patterns, as shown in Figure 1(d). In [8, 13, 21], the data of utility fraud users is synthetic, which is too simple to capture actual gas-theft behaviors of restaurant users. Moreover, as shown in Figure 1(b), utility fluctuations are inevitable in real life, which are likely to raise false alarms by general methods of time series anomaly detection [17, 24]. Because these methods are usually sensitive to point-wise anomalies.

In this paper, we propose a gas-theft detection method *msRank* to discover suspicious restaurant users when only scarce labels are available. Our method contains three main components: 1) *data pre-processing*, which filters reading noises and excludes data-missing or zero-use users; 2) *normal user modeling*, which quantifies the self-stable seasonality of normal users and distinguishes them from unstable ones; and 3) *gas-theft suspect detection*, which discovers gas-theft suspects among unstable users by RankNet-based suspicion scoring on the extracted deviation features. The component of normal user modeling and that of gas-theft suspect detection are connected seamlessly to overcome the label scarcity. By modeling normal users, we not only decrease the overall complexity by narrowing the suspect scope, but also provide normal users as negative samples. The suspect detection then adopts normal users along with gas-theft labels to construct positive-negative sample pairs for training RankNet. Our contributions are four folds:

- We provide the first attempt to detect gas-theft restaurant users by mining gas consumption data. And we overcome the problem of label scarcity which commonly exists in industrial scenarios.
- We propose a normal user modeling module to quantify the self-stable seasonality of normal users and distinguishes them from unstable ones. We also propose a gas-theft suspect detection module to discover suspicious users among unstable ones by RankNet-based suspicion scoring.
- We evaluate our *msRank* extensively on three real-world datasets of gas consumption. Experiment results demonstrate the advantages of our method which outperforms the best baseline by 25% as for the top 10% hit rate.
- We have deployed a system *GasShield*, providing the gas-theft suspect list weekly for a gas group in northern China. Gas thefts can thus be discovered in the early stage with higher accuracy.

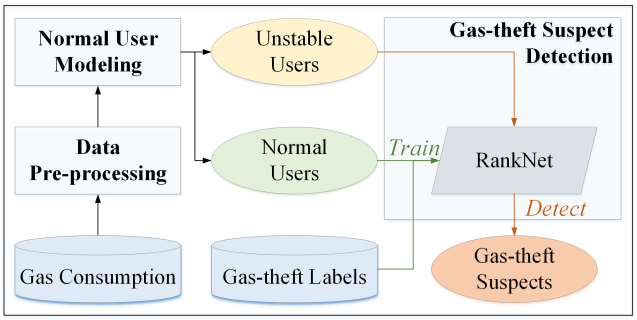


Figure 2: Framework of *msRank*.

2 OVERVIEW

2.1 Problem Definition

Given hourly gas consumption records $C_i = (c_i^1, c_i^2, ..., c_i^T)$ *of historical* T *time steps for each restaurant user* $u_i \in \boldsymbol{U}$*, we aim to detect whether the user has gas theft behaviors.*

2.2 Framework

Based on expert experience, gas theft means cause mainly three types of anomalies in gas consumption data: 1) missing data, 2) zero use, and 3) irregular patterns. The first two can be easily identified, while the third one is quite hard to detect. Therefore, we detect data-missing or zero-use users in data pre-processing, and then handle the third gas-theft behaviors by a two-step solution. The framework of our method *msRank* is illustrated in Figure 2.

Data Pre-processing. This component takes raw data and performs three tasks: 1) *Noise Filtering*, which removes readings of unrealistic magnitudes; 2) *Data-missing User Exclusion*, which excludes users of high data miss rate; 3) *Zero-use User Exclusion*, which excludes users of high data zero rate.

Normal User Modeling. This component takes pre-processed users with cleaned data and performs two tasks: 1) mode sequence construction, which converts gas consumption time series into bi-seasonal mode sequences; and 2) entropy-based stability detection, which calculates the mode entropy *ModeEn* based on mode sequences, and then distinguishes normal users from unstable ones according to *ModeEn*.

Gas-theft Suspect Detection. This component extracts finer -grained features to portray the deviation degree of gas consumption from intrinsic self-stability, and ranks unstable users detected in the previous step based on their gas-theft suspicion scores. The suspicion score is predicted by a RankNet model, which is trained with gas-theft labels and detected normal users. Users are regarded as gas-theft suspects if their scores are above a threshold determined by the percentage of potential gas thefts provided by experts.

3 DATA PRE-PROCESSING

In this section, we pre-process gas consumption data by cleaning noises in raw readings and excluding data-missing or zero-use users. As shown in Figure 3, the two types of users take 38% and 16% respectively among gas-theft labels, indicating that such data quality problems can potentially reflect gas theft behaviors.

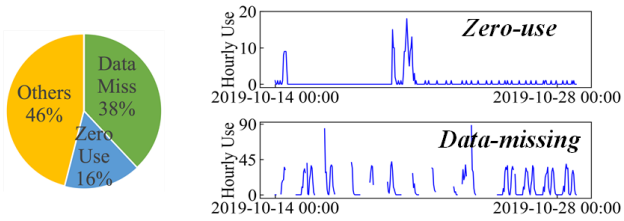


Figure 3: Anomaly type distribution of gas-theft labels.

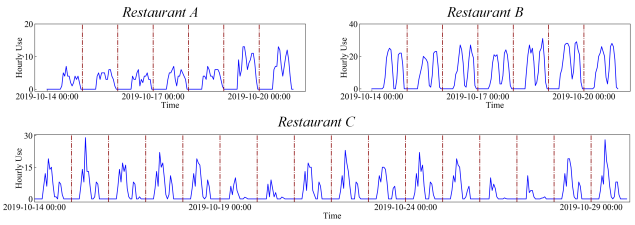
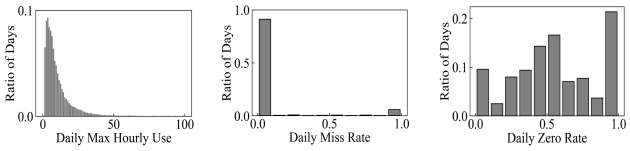


Figure 5: Diversity of normal gas consumption behaviors.

Noise Filtering. For restaurant users, the scale of gas consumption volume depends on the gas-burning appliances they use, which are mostly cookers. As Figure 4(a) shows, daily max hourly uses of restaurants rarely exceed $100m^3$. However, some faulted gas meters may record extremely large values out of realistic magnitudes. So we remove hourly gas consumption readings larger than 100.

Data-missing User Exclusion. Gas-theft users usually break gas equipment, which can cause the collected data missing frequently. As Figure 4(b) shows, for the dominant majority of restaurants, their daily miss rates are lower than 5%, while a tiny part of users lack more than 90% data in some days. So we exclude users with daily miss rates higher than 25% for more than 7 days.

Zero-use User Exclusion. Readings containing many zero values indicate that the restaurant seldom uses gas or steals gas by forcing meters to stop recording. As Figure 4(c) shows, daily zero rates are mainly lower than 90% and concentrate around 50%, which is consistent with opening hours of restaurants. However, some users have daily zero rates higher than 90%. So we exclude users with daily zero rates higher than 90% for more than 7 days.



(a) Daily Max Hourly Use (b) Daily Miss Rate (c) Daily Zero Rate

Figure 4: Distribution of data quality metrics.

After data pre-processing, remaining restaurant users with data of higher quality will be analyzed in following components.

4 NORMAL USER MODELING

In this section, with the data cleaned in Sec. 3, we aim to find normal users with high confidence which take the majority, and to tell these normal users apart from unstable users which include gas-theft suspects. So that we not only decrease the overall complexity by narrowing the scope of suspects, but also provide negative samples for the gas-theft suspect detection later.

Challenges. With dozens of normal users provided by experts as shown in Figure 5, we observe that bi-seasonal patterns repeat steadily in their hourly gas consumption, where the daily patterns are nested in the weekly ones. However, it is non-trivial to model such self-stability of normal users due to two reasons:

- Normal users exhibit diverse patterns, while few of them have been labeled by on-site inspections in the past. So it is impractical to do classification directly with such scarce normal labels.
- Utility fluctuations, like amplitude shifts, spikes and dips, are inevitable in real life and will disable conventional measures. For example, the Autocorrelation Coefficient (ACF) [11], widely used to quantify the self-similarity of periodic time series, can be affected by such fluctuations. The three users in Figure 5 are all self-stable, while given their gas consumption of the same period, the ACFs of *Restaurant A,C* are evidently lower than that of *B*.

Insights. We propose to find high-confidence normal users by examining their self-stability of gas consumption modes, inspired by our observations from data. Though the three restaurants in Fig. 5 have different gas consumption behaviors and the normal user in Fig. 1 appears occasional fluctuations colored in orange, certain seasonal patterns repeat steadily for each of them. It is consistent with the common sense that the gas consumption patterns of a restaurant is decided by its business mode, and is thus fixed.

Main Idea. To characterize the self-stability of gas consumption patterns for each restaurant, and to identify normal users, the procedure of normal user modeling is devised as shown in Figure 6, which consists of two main steps: 1) *Mode Sequence Construction*, which first generates seasonal gas consumption patterns for each user, then clusters patterns of all users to discover gas consumption modes that commonly exist, and finally converts gas consumption time series into mode sequences for each user based on the discovered modes; 2) *Entropy-based Stability Detection*, which calculates the mode entropy *ModeEn* of each user based on its mode sequences, and identify normal users according to it.

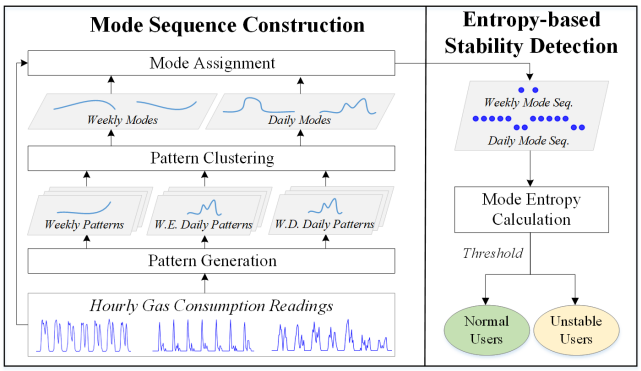


Figure 6: Normal user modeling.

4.1 Mode Sequence Construction

In this step, we want to convert time series of hourly gas consumption into bi-seasonal mode sequences. So that the generated sequences can be insensitive to utility fluctuations and focus on the key information of pattern self-stability, using which we detect normal users further.

To avoid being disturbed by utility fluctuations and reading noises, we should make the mode sequences coarse-grained, where each mode represents a typical daily or weekly gas consumption behavior. An intuitive idea is to segment raw time series by day and by week, and then cluster them to find intra-user modes. However, recall that we can only use the data of limited length to detect whether the user is a gas theft suspect. For each user, its seasonal segments are too limited to discover general modes, which makes it difficult to judge individually whether they are reasonable.

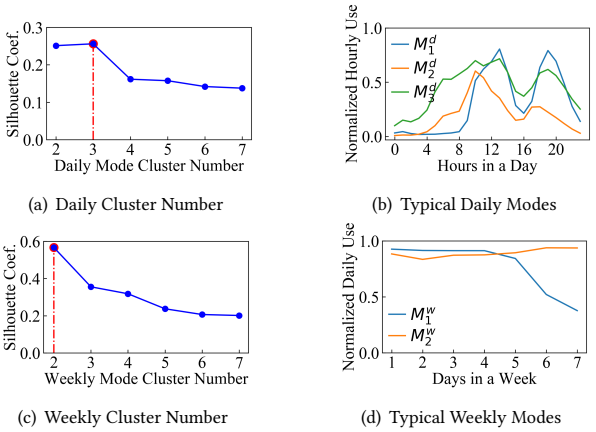
Therefore, the main idea of mode sequence construction is that we first discover daily and weekly modes from behaviors of all users; and then for each user, we generate the mode sequence based on the widely-shared typical gas consumption modes. It mainly contains three steps: 1) *Pattern Generation*, which generates the bi-seasonal gas consumption patterns of each user; 2) *Pattern Clustering*, which clusters patterns of all users, and regards cluster centroids as typical gas consumption modes; and 3) *Mode Assignment*, which converts gas consumption time series into bi-seasonal mode sequences based on the discovered daily and weekly modes.

Pattern Generation. As described above, a normal user can still have fluctuating records. To obtain robust gas consumption modes, we generate gas consumption patterns of each user (i.e., each clustering sample) by averaging its gas consumption. Specifically, we generate the *daily pattern* and the *weekly pattern* for each restaurant user u_i. Because gas consumption behaviors might be different on workdays and on weekends, the *daily pattern* is further divided into the *workday pattern* and the *weekend pattern*:

- Daily pattern of workdays $\mathbf{P}_i^{wd} \in \mathbb{R}^{24}$, which is the averaged hourly gas consumption over workdays for a given user.
- Daily pattern of weekends $\mathbf{P}_i^{we} \in \mathbb{R}^{24}$, which is the averaged hourly gas consumption over weekends for a given user.
- Weekly pattern $\mathbf{P}_i^{w} \in \mathbb{R}^{7}$, which is the averaged daily gas consumption over days in a week for a given user.

The generated patterns characterize intrinsically the gas consumption behaviors of each restaurant. However, two restaurants of the same business mode can still be dissimilar in the volume of gas consumption. To remove the amplitude differences, we normalize all patterns using the min-max strategy.

Pattern Clustering. In this step, we cluster the generated and normalized patterns of all users, to obtain gas consumption modes based on cluster centroids. As illustrated in Figure 6, the workday daily patterns $\mathbf{P}^{wd} = \{\mathbf{P}_i^{wd} | \forall u_i \in \mathbf{U}\}$ and the weekend daily patterns $\mathbf{P}^{we} = \{\mathbf{P}_i^{we} | \forall u_i \in \mathbf{U}\}$ are clustered together to discover daily modes; the weekly patterns $\mathbf{P}^{w} = \{\mathbf{P}_i^{w} | \forall u_i \in \mathbf{U}\}$ are clustered to discover weekly modes. Since the gas consumption modes are limited considering Chinese dietary habits, kMeans clustering [15] is adopted, where the cluster number k^d and k^w of daily and weekly modes are determined according to silhouette coefficients (SC) [18].



(a) Daily Cluster Number (b) Typical Daily Modes

(c) Weekly Cluster Number (d) Typical Weekly Modes

Figure 7: Illustration of pattern clustering.

The SCs are the best when $k_d = 3$ and $k_w = 2$ as shown in Figure 7(a) and 7(c) respectively. The obtained cluster centroids are the gas consumption modes that we look for.

Figure 7(b) shows the three corresponding daily modes:

- M_1^d: They supply both lunches and dinners but no breakfast, like restaurants of hot pots which are too spicy for breakfast.
- M_2^d: They supply mainly breakfast with minor traffic appearing near the evening peak, like some snack stands selling steamed buns, dumplings, noodles, etc.
- M_3^d: They supply three meals in a day, like Cantonese morning tea restaurants, fast food restaurants, etc.

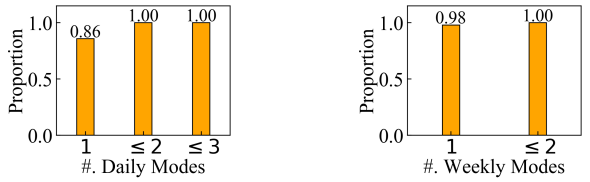
Figure 7(d) shows the two corresponding weekly modes:

- M_1^w: They mainly supply working meals, like restaurants nearby office buildings where the traffic drops sharply on weekends.
- M_2^w: Their traffic is similar in a week, with slightly increasing on weekends perhaps also on Friday. Since people may prefer to dine out for leisure on days off.

Mode Assignment. In this step, we construct daily mode sequences and weekly mode sequences using the gas consumption modes discovered above. The main idea is that, for each user, we first segment the time series of hourly gas consumption by the day, and then assign each slice to its nearest daily mode based on the Euclidean distance. As shown in Figure 6, we obtain the daily mode sequence $\mathbf{M}_i^d =< m_1^d, m_2^d, \cdots, m_D^d >$, where $m_p^d \in [1, ..., k^d]$ and D is the total of days. Similarly, the weekly pattern sequence $\mathbf{M}_i^w =< m_1^w, m_2^w, \cdots, m_W^w >$ are generated from the daily gas consumption, where $m_q^w \in [1, ..., k^w]$ and W is the total of weeks.

4.2 Entropy-based Stability Detection

In this step, we aim to quantify the self-stability of mode sequences, in order to distinguish normal users from unstable ones. Since normal restaurants exhibit stable modes of gas consumption, while patterns of gas-theft users are chaotic and random, it is natural to devise an entropy-based method to quantify the self-stability of mode sequences. Therefore, we first define an entropy-based method to quantify behaviors of stable users based on their generated daily and weekly mode sequences, and then regard users whose mode entropy is below a specific threshold as normal ones.



(a) #. Daily Mode Distribution (b) #. Weekly Mode Distribution

Figure 8: Insight of designing *ModeEn*.

Given the daily mode sequence of a user $\mathbf{M}_i^d$, the entropy H in Eq.1 quantifies how chaotic the mode sequence is. The more consistent modes $\mathbf{M}_i^d$ contains, the smaller $H(\mathbf{M}_i^d)$ will be.

$$H(\mathbf{M}_i^d) = -\sum_{x=1}^{k^d} p_x ln(p_x) \tag{1}$$

where p_x is the occurrence proportion of daily mode k_x^d in $\mathbf{M}_i^d$.

However, such entropy is insufficient to capture the stability of restaurant users. We say a user has n daily gas consumption modes if his/her most frequent n daily modes in $\mathbf{M}_i^d$ take more than 75% in the sequence, where n is the minimum number of modes that satisfies the condition. According to Figure 8(a), users with single daily mode can cover only 86% restaurants, while the percentage of gas-theft users are much smaller according to the domain knowledge. If we apply the entropy directly to quantify the mode stability of users, many normal users would be omitted.

Considering that, we design the mode entropy *ModeEn*, which treats not only users of single daily mode as normal ones, but also those of double daily modes with single weekly mode at the same time. This is inspired by two observations from data:

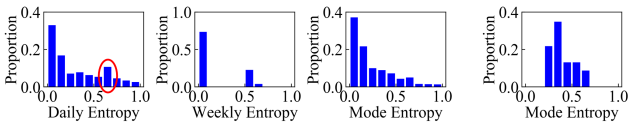
- *For the majority of users, they have at most two daily gas consumption modes.* As shown in Figure 8(a), only 86% users exhibit single daily mode. But if we jointly consider single-mode and double-mode users, all users can be covered, which implies that they normally have only one or two daily modes.
- *For the majority of users, they have only one weekly gas consumption mode.* As shown in Figure 8(b), regardless of the daily mode number, almost all users have only one weekly mode.

To capture the aforementioned two characteristics, the mode entropy *ModeEn* is defined as follows:

$$ModeEn(\mathbf{M}_i^d, \mathbf{M}_i^w) = H(\mathbf{M}_i^d) - \alpha \cdot \mathbb{1}(H(\mathbf{M}_i^w) == 0 \& H(\mathbf{M}_i^d) > \alpha) \tag{2}$$

where $\mathbb{1}(\cdot)$ is an indicator function which returns 1 if the condition holds, and 0 otherwise. And α is an entropy shifting parameter.

The *ModeEn* is able to quantify the stability of both single daily mode users and double daily mode users with stable weekly mode.



(a) Composition of *ModeEn* (b) *ModeEn* of labels

Figure 9: Distribution of *ModeEn*.

For single mode users, their $H(\mathbf{M}_i^d)$ are small, and the entropy shift will not be activated. For double mode users, their $H(\mathbf{M}_i^d)$ are relatively large, since $\mathbf{M}_i^d$ contains 2 modes mainly. And the entropy shifting parameter α will be applied only if $H(\mathbf{M}_i^w) = 0$, i.e., the weekly mode is totally stable. In this way, for both single and double daily mode users, *ModeEn* will be small if they are stable.

The entropy shifting parameter α is set equal to the entropy of a daily mode sequence which is exactly binary for being workday-or-not over its time span. And we only shift the entropy for mode sequences in which the double daily modes follow the workday-or-not pattern. For example, in our evaluation dataset, there are 19 workdays and 12 days-off (including the Chinese National Day) in October 2019. Therefore, α is set to $-\frac{19}{31}\ln\frac{19}{31} - \frac{12}{31}\ln\frac{12}{31} = 0.67$.

Figure 9(a) shows the distribution of *ModeEn* and its components. It can be noticed that $H(\mathbf{M}_i^d)$ are mostly close to 0 while a peak appears near 0.6-0.7. That indicates restaurants normally exhibit single daily pattern, while some may have daily patterns switched regularly between workdays and days-off. The majority of $H(\mathbf{M}_i^w)$ are close to 0, showing that most restaurants exhibit single weekly pattern. As for *ModeEn*, its distribution on all users contains one dominant peak close to 0, distinct from that on labels shown in 9(b).

After *ModeEn* being calculated for each user, users with *ModeEn* below a specific threshold β are treated as normal ones. β is selected from the sharp decrease point in *ModeEn* distribution of all users.

5 GAS-THEFT SUSPECT DETECTION

With Section 4, normal users are identified, and there remain unstable users to be analyzed further. In this section, we aim to detect gas-theft suspects among unstable users. Because users who don't steal gas but have occasionally fluctuating modes can also be regarded as unstable users based on their *ModeEn*.

Challenges. With *ModeEn*, though it is clear to detect normal users, its distribution on gas-theft labels is overlapped with that on most unstable users as Figure 9(b) shows. So we can't distinguish gas-theft suspects further with merely the information above.

Main Idea. To achieve our goal, we employ a ranking-based model, i.e., RankNet, to score the suspicion level of each unstable user. We feed RankNet with finer-grained features describing to what extent users' gas consumption behaviors deviate from their intrinsic seasonal patterns. We train RankNet with positive-negative sample pairs of gas-theft labels and normal users, and then use the trained model to predict on unstable users. In this way, we establish a comparative relation that suspicion scores of gas-theft labels are all higher than those of normal users. So given an unstable user, the higher its score is, the more suspicious it is detected to be.

5.1 Feature Extraction

We extract two categories of finer-grained features from hourly gas consumption data, making it more distinct that to what extent users' gas consumption behaviors deviate from their intrinsic seasonality.

Daily Deviation Features. For each restaurant user, to quantify the deviation degree of gas consumption C_i from its intrinsic seasonality, we first generate a daily deviation sequence $\mathbf{V}_i =< v_1, v_2, \cdots, v_D >$ with D the total of days, and then extract features from $\mathbf{V}_i$ as listed in Table 1, which portray the distribution of daily

Table 1: Extracted features for RankNet.

Category	Feature Description	Dimension
Daily Deviation Features	$Q1, Q2, Q3$, 0.9-quantile of V	4
	Mean Absolute Deviation of V	1
	Mean of V	1
	STD of V	1
ACF Features	$ACF_k, k = 24, 120, 168$	3

deviation degrees. $\mathbf{V}_i$ is generated as follows: 1) STL decomposition [5] on C_i, extracting its seasonal component S_i as the baseline; 2) Min-max normalization on both C_i and S_i; 3) Calculating the Pearson's distance [10] between normalized C_i and S_i per day.

Autocorrelation Features. Autocorrelation Coefficient (ACF) [11] represents the correlation among values of the same observation at different times. It is widely used to find repetitive patterns in periodic but probably noisy time series. Since we already know that there should exist both daily and weekly patterns in hourly gas consumption C_i, we calculate the ACF_k of C_i with the time lags $k = 24, 120, 168$ respectively, in order to measure its self-similarity at 1-, 5-(length of weekdays) and 7-day intervals.

5.2 RankNet-based Suspect Detection

In this step, based on the aforementioned features extracted from hourly gas consumption data, we score each user using the RankNet model and report gas-theft suspects among unstable users.

The structure of RankNet is shown in Figure 10. RankNet takes various features $\mathbf{x}$ extracted in Sec. 5.1, and generates a score $s = f(\mathbf{x}; \mathbf{w})$ through the neural network with the learned parameter $\mathbf{w}$.

During the training phase, a positive-negative sample pair (i.e., a gas-theft label and a detected normal user) is sent into RankNet each time, and the learning procedure is trying to score the gas-theft label higher than the normal one. When we construct the training pair u_i and u_j, if u_i is a gas-theft user and u_j is a detected normal user, the label is 1, otherwise 0. The probability P_{ij} that u_i is more suspicious than u_j is modeled as $P_{ij} = \sigma(s_i - s_j)$, where $s_i = f(\mathbf{x}_i; \mathbf{w})$, $s_j = f(\mathbf{x}_j; \mathbf{w})$, $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{x}_i$, $\mathbf{x}_j$ are features extracted from u_i and u_j respectively. Therefore, $\mathbf{w}$ can be easily learned through the gradient descent by minimizing the cross entropy loss between P_{ij} and the label.

During the inference phase, we send features extracted from unstable users into the network, and the users with higher scores are more suspicious. To provide target users for on-site inspections, we can report top N% suspicious users as gas-theft suspects.

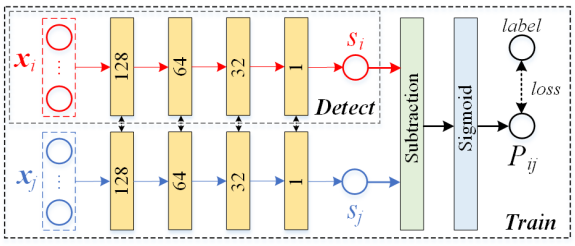


Figure 10: Structure of RankNet.

6 EXPERIMENTS

6.1 Experimental Settings

Datasets. We conduct experiments on three real-world datasets collected by three branches of a gas group, denoted by company *A*, *B*, and *C* for short, which cover different districts of one city in China. The statistics details of all restaurant users and users remained after data pre-processing is shown in Table 2, where there are only 23 users labeled as gas thefts. Each restaurant user has a time series of its hourly gas consumption, of which the time span lasts from Oct. 1st, 2019 to Oct. 31th, 2019.

Table 2: Details of datasets (#. Restaurants)

Dataset			*A*	*B*	*C*
Raw			4,282	2,306	1,320
After D.P.	Unlabeled	Normal	1,329	838	418
		Unstable	961	554	326
	Labeled Gas Thefts		11	8	4

Evaluation Methods. We adopt two of the three datasets after data pre-processing as the training set, and the left one for evaluating the gas-theft suspect detection. Moreover, to overcome the randomness, the evaluation is repeated for five times, of which the average performance is reported.

We use the *HitRate@N%* as our evaluation metric, which is defined as the ratio between the number of gas-theft labels contained in the top N% suspicious users detected by algorithms and the number of all labels in the evaluation dataset. We report the hit rates in top 5%, 10%, and 15% suspicious users.

$$HitRate@N\% = \frac{\#positive\ labels\ in\ topN\%\ suspicious\ users}{\#positive\ labels} \quad (3)$$

We note that conventional metrics, e.g., precision, recall, (best) F-score and AUC, are not applicable in our task. Since they require a dataset fully labeled for both normal users and abnormal users [4, 8, 13, 17], which are difficult to obtain in industrial scenarios.

Baselines. We compare our *msRank* with four baselines. The first two baselines are unsupervised, which are directly applied to the restaurant users after data pre-processing. The next two baselines are supervised, for which gas-theft labels are used as positive samples and unlabeled users after data pre-processing as negative ones.

- **LOF** [2]: *Local Outlier Factor* is unsupervised, which measures the local density deviation of given samples and detects whether they have lower densities compared to their neighbors.
- **DAGMM** [26]: *DAGMM* is an unsupervised anomaly detection method for high-dimensional data, which combines the deep auto-encoder and the Gaussian mixture model.
- **RankNet** [3]: *RankNet* as a baseline is trained directly using features elaborated in Section 5.1.
- **SR-CNN** [17]: *SR-CNN* is a state-of-the-art method for time series anomaly detection, which uses Spectral Residual to amplify anomaly points in time series and is trained end to end.

Variants. We also compare *msRank* with its four variants:

- **DBRank**: *DBRank* uses the DBSCAN clustering for normal user modeling, where $minPts = 4$ and $\epsilon = 1.6$. Users that are not

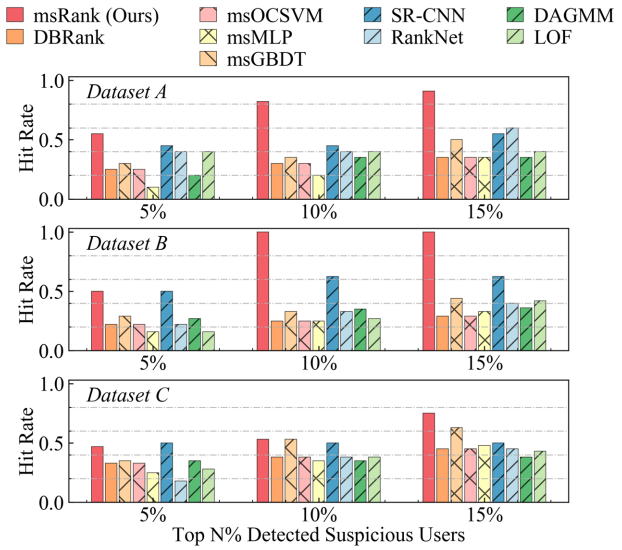


Figure 11: Effectiveness evaluation.

outliers are treated as normal samples and fed into the module of gas-theft suspect detection.

- **msOCSVM**: *msOCSVM* uses the normal user modeling to find normal users, and trains one-class SVM [19] with them. Unstable users are ranked by the probability the trained OCSVM predicts.
- **msMLP**: *msMLP* uses the normal user modeling to find normal users, and trains MLP with gas-theft labels and them. Unstable users are ranked by the probability the trained MLP predicts.
- **msGBDT**: *msGBDT* is similar to the above one. The difference is that MLP is replaced with the GBDT [9] classifier.

Parameter Setting. The entropy shifting parameter α and the *ModeEn* threshold β is set as 0.67 and 0.2 respectively as stated in Sec. 4.2. The number of inferred normal and unstable users are shown in Tab. 2. RankNet employed in gas-theft suspect detection contains three hidden layers with 128, 64, and 32 hidden units respectively. Each unit uses the ReLu as the activation function.

Implementation. Our algorithms are implemented with Keras. Experiments are conducted on a workstation with an Intel(R) Core(TM) CPU i7-8700K @ 3.7GHz, 32GB memory, and Windows 10 OS.

6.2 Effectiveness Evaluation

Overall Evaluation. We first compare *msRank* with four baselines under the similar condition that gas-theft labels are scarce, as shown in Figure 11. LOF, which calculates the density based on distances, does not perform well on high-dimensional time series. Though DAGMM can be used on the unsupervised situation, it is more effective when negative samples are available. Therefore, it neither shows a good performance. RankNet and SR-CNN both leverage the gas-theft labels which make their hit rates higher. Our *msRank* outperforms all baselines in most cases, since it not only leverages positive labels but also detects negative labels (normal users) to improve the accuracy of suspicion scoring.

Effectiveness of Normal User Modeling. We demonstrate the effectiveness of the normal user modeling component by comparing *msRank* with RankNet and DBRank, as shown in Figure 11. With normal user modeling, our *msRank* performs evidently better than RankNet. The reason is that, the negative samples we feed to the model are more accurate when they are detected normal users instead of unknown users. The comparison between *msRank* and DBRank shows that, normal user modeling based on the *ModeEn* is much more effective than simply clustering gas consumption time series in terms of finding normal users.

Effectiveness of RankNet. With the module of normal user modeling providing negative samples, three well-performed supervised models commonly used in utility fraud detection can be employed. We compare *msRank* with msOCSVM, msMLP and msGBDT to show the effectiveness of ranking-based detection, as shown in Figure 11. *msRank* performs the best among these classification-based models. There are two reasons behind it: 1) Actually, we know few real negative samples. The normal users detected by normal user modeling may contain noises, which makes it difficult to train classification-based models well. While the ranking-based method focuses on modeling the relative relationship, which is more robust; and 2) The pair-wise training method generates more training samples, thus the data we have can be fully leveraged.

7 SYSTEM DEPLOYMENT

Our system, i.e., *GasShield*, is deployed in a gas group in northern China, and used internally to monitor real-time gas consumption anomalies. The backend is implemented using the Flask and MySQL, and the frontend is written using jQuery, Bootstrap and ECharts. The gas consumption records collected by meters are reported to Hive, and the data migrates to our MySQL database using Sqoop weekly. After data pre-processing, *msRank* is called weekly to predict the gas-theft suspicion score for each restaurant user, based on readings of the last 30 days.

The system interface of *GasShield* is shown in Figure 12. It mainly consists of three panels: 1) *User Anomaly Type Distribution*, which gives the anomaly type distribution based on weekly predicted results; 2) *Extremely Suspicious Users*, which lists the most suspicious users (Top 5%) we detected in the descending order, so that operators can conduct more targeted on-site inspections; and 3) *Hourly Gas Consumption*, which displays the hourly gas consumption records during the last 30 days for a selected user.

Based on both the incoming and the historical gas consumption data, the threshold β for normal user modeling will be renewed and the RankNet will be re-trained with the newly inspected normal users as well as gas-theft users every 30 days.

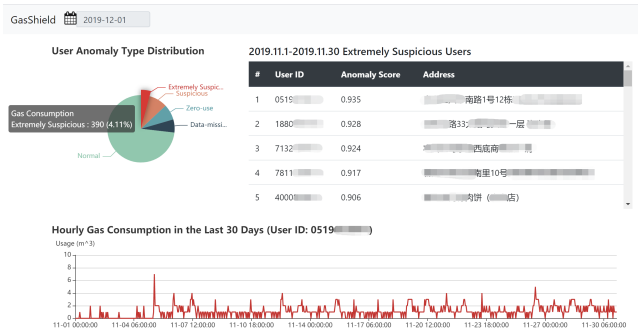


Figure 12: User interface of *GasShield* system.

8 RELATED WORK

8.1 Gas Theft Detection

Traditional methods of gas theft detection highly rely on active human efforts [25], which are costly yet ineffective. Hardware solutions add protective devices onto meters [23], [16] proposes to find gas-theft users using statistical indicators, while they both show little effect actually. Different from them, we propose a data-driven method *msRank* based on mining massive gas consumption data, which greatly increases the efficiency of gas theft detection.

8.2 Utility Fraud Detection

Utility fraud is a worldwide concern for energy suppliers (gas, power, and water). Plenty of data-driven methods, supervised or semi-supervised, are proposed to detect it. They require fully labeled data either synthetic or collected from numerous on-site inspections [6, 7]. Classification-based methods are widely adopted, which are summarized in [4]. Clustering-based [21] is trained on normal samples and tested on synthetic anomalies. Prediction-based [8] models normal behaviors so that deviations (synthetic noises) are detected as anomalies. Semi-supervised [13] first extracts features under semi-supervision and then detects frauds in a supervised way. To the best of our knowledge, our *msRank* is the first method of utility fraud detection based on scarcely labeled real-world data, which achieves superior performance proven by experiments.

8.3 Time Series Anomaly Detection

To detect gas-theft suspects based on gas consumption data is also a problem of time series anomaly detection, for which existent methods present promising effects. They are mainly statistical [12, 20, 22], supervised [14] or unsupervised [17, 24]. Aiming to monitor service metrics, they mostly focus on point-wise anomalies like spikes and dips, amplitude shifts, etc. However, utility fluctuations are inevitable in real life. These methods can report many false alarms in our scenario. Instead, our *msRank* detects gas-theft suspects by capturing specific gas consumption patterns of restaurant users.

9 CONCLUSION

In this paper, we propose *msRank* to detect gas-theft restaurant users with gas consumption data, overcoming the issue of label scarcity. *msRank* first tells normal and unstable users apart based on their gas consumption mode stability, which is quantified by *ModeEn*. Then, a RankNet-based suspect detection method ranks unstable users by their suspicion levels, providing suggestions to inspectors. Extensive experiments on three real-world datasets show its effectiveness, and *msRank* outperforms the best baseline by 25% in HitRate@10%. A system based on *msRank*, i.e., *GasShield*, is deployed and used internally by a gas group of one city in northern China. In the future, we will focus on generalizing our method to more types of gas users and other utility fraud detection tasks.

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China (2019YFB2101801), the National Natural Science Foundation of China (No. 61773324), China Postdoctoral Science Foundation, and Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] China Hospitality Association. 2019. 2019 China Restaurant Industry Survey Report. https://bit.ly/3cbeYwd.
[2] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.
[3] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *ICML*. 89–96.
[4] Madalina Mihaela Buzau, Javier Tejedor-Aguilera, Pedro Cruz-Romero, and Antonio Gómez-Expósito. 2018. Detection of non-technical losses using smart meter data and supervised learning. *IEEE Transactions on Smart Grid* 10, 3 (2018), 2661–2670.
[5] Robert B Cleveland et al. [n.d.]. STL: A seasonal-trend decomposition procedure based on loess. 1990. *DOI: citeulike-article-id* 1435502 ([n. d.]).
[6] Bernat Coma-Puig, Josep Carmona, Ricard Gavalda, Santiago Alcoverro, and Victor Martin. 2016. Fraud detection in energy consumption: A supervised approach. In *DSAA*. IEEE, 120–129.
[7] Breno C Costa, Bruno LA Alberto, André M Portela, W Maduro, and Esdras O Eler. 2013. Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process. *International Journal of Artificial Intelligence & Applications* 4, 6 (2013), 17.
[8] Vitaly Ford, Ambareen Siraj, and William Eberle. 2014. Smart grid energy fraud detection using artificial neural networks. In *CIASG*. IEEE, 1–6.
[9] Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational Statistics Data Analysis* 38, 4 (2002), 367–378.
[10] MH Fulekar. 2009. *Bioinformatics: applications in life and environmental sciences*. Springer Science & Business Media.
[11] John A Gubner. 2006. *Probability and random processes for electrical and computer engineers*. Cambridge University Press.
[12] Jordan Hochenbaum, Owen S Vallis, and Arun Kejariwal. 2017. Automatic anomaly detection in the cloud via statistical learning. *arXiv preprint arXiv:1704.07706* (2017).
[13] Tianyu Hu, Qinglai Guo, Xinwei Shen, Hongbin Sun, Rongli Wu, and Haoning Xi. 2019. Utilizing unlabeled data to detect electricity fraud in AMI: A semisupervised deep learning approach. *IEEE transactions on neural networks and learning systems* 30, 11 (2019), 3287–3299.
[14] Dapeng Liu, Youjian Zhao, Haowen Xu, Yongqian Sun, Dan Pei, Jiao Luo, Xiaowei Jing, and Mei Feng. 2015. Opprentice: Towards practical and automatic anomaly detection through machine learning. In *Proceedings of the 2015 Internet Measurement Conference*. 211–224.
[15] J. Macqueen. 1967. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 281–297.
[16] Li Manman, Gu Xiaopeng, and Wuteng Yang. 2018. Study and Verification on a Query Method of Abnormal Gas Consumption Data of Restaurant Users. *ChengShi RanQi* 12 (1 2018), 19–21.
[17] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-Series Anomaly Detection Service at Microsoft. In *KDD*. 3009–3017.
[18] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.
[19] Bernhard Scholkopf, Alexander J Smola, Robert C Williamson, and Peter L Bartlett. 2000. New Support Vector Algorithms. *Neural Computation* 12, 5 (2000), 1207–1245.
[20] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. 2017. Anomaly detection in streams with extreme value theory. In *KDD*. 1067–1075.
[21] Joaquim L Viegas and Susana M Vieira. 2017. Clustering-based novelty detection to uncover electricity theft. In *FUZZ-IEEE*. IEEE, 1–6.
[22] Li Wei, Nitin Kumar, Venkata Nishanth Lolla, Eamonn J Keogh, Stefano Lonardi, and Chotirat (Ann) Ratanamahatana. 2005. Assumption-Free Anomaly Detection in Time Series.. In *SSDBM*, Vol. 5. 237–242.
[23] Zhang Xiongjun, Liu Xingwei, and Geyang Zhou. 2020. Study on Gas Theft and Governance Abroad. *Gas Heat* 40, 4 (5 2020), 37–40.
[24] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. 2018. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. In *WWW*. International World Wide Web Conferences Steering Committee, 187–196.
[25] Lijuan Zhou and Gao Tianyu. 2018. A Brief Talk on the Status Quo and Problems of Governing theft in Beijing Gas Group. *ChengShi RanQi* 11 (12 2018), 26–29.
[26] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep Autoencoding Gaussian Mixture Model For Unsupervised Anomaly Detection. (2018).