Identifying ATM Fraud Transactions in Thailand using Outlier Detection with Location-Based Grouping and Behavior Feature

Natsuda Kaothanthong[1] and Roongtawan Laimek[2]

[1,2]Sirindhorn International Institute of Technology, Thamamsat University,
Pathumthani, Thailand
Email: natsuda@siit.tu.ac.th, amabird@live.com

Received: October 27, 2021 / Reviesed: January 31, 2022 / Accepted: February 21, 2022

***Abstract*—Financial fraud causes a major loss to a bank. The challenge of classifying fraud is a high true positive rate while keeping the number of false positives as low as possible. One difficulty is the unbalanced size of the labeled data which causes a low detection rate and a high false-positive rate. We present a method to sample the data to cope with the unbalance problem in the fraud detection problem. The location feature is applied to separate accounts into 'local-only' and 'has-abroad'. The proposed feature extraction can separate many fraud transactions from legitimate transactions. To differentiate fraud from legitimate transactions, fraud can be considered an outlier. Transformation functions, deviation, risk, and probability features are applied in this work to both numeric and non-numeric features. The experimental result shows that the location-based separation together with the proposed features achieves higher TRP and lower FPR than not dividing the group. It achieves a true positive rate of 75.00% for 'local-only' and 100% for 'has-abroad'. The lowest false positive rate is 8.23% for 'has-abroad'. Comparing the efficiency of the proposed features with the classification using an isolation forest, the true positive rate is improved from 56.25% to 75.00% and the false positive rate is increased from 2.47% to 28.02%.**

***Index Terms*—ATM Fraud, Outlier Detection**

I. Introduction

Bank fraud is a deceptive activity for monetary gain. Frauds involve many forms of complicated financial transactions depending on the tools that a fraudster is targeting. According to a survey by [1], 23% of respondents in Thailand encountered fraud in 2016. An Automatic Teller Machine (ATM) is widely used for many purposes, i.e., withdrawing, transferring, and making bill payments. In addition, a credit card and a debit card can be used for withdrawing money from an ATM, which can be targeted by fraudsters. A skimming technique can be used to steal card information and duplicate it into white fake cards, to be used as an identification of the account for a cash withdrawal.

Fraud detection is implemented to identify a suspicious transaction, to prevent losses to the bank. Although machine learning has extensively been studied for credit card fraud detection, the prediction model cannot directly be implemented for an ATM because the channel of transactions affects the different fraud behaviors. Formerly, it was manually performed by fraud analysts. Each ATM transaction was inspected by rule-based software and the suspicious transactions were notified to the analyst. The fraud investigation was taken manually by making a phone call to the account's owner for a confirmation of the transaction. As the number of transactions has increased extensively and many false-positive transactions were reported by the rule-based software, such a method becomes unfeasible. Machine learning utilizes previously known fraud transactions to construct a model to predict a new transaction. The suspicious transaction can then be notified automatically.

The ATM fraud detection models can be summarized into three categories: (1) Aggregation [2]-[4], (2) Binary Classification [5]-[11], and (3) Outlier Detection [12], [13], [14]. The aggregation method extracts the pattern of transactions. The numeric features in a fixed period are aggregated to represent a normal behavior [4]. In addition, [3] defined the behavior pattern using non-numeric values such as point-of-sale terminals and type of transactions to define the risk of the transaction. The usage of a credit card was identified as a sequence of operations [2]. The relationship among the transactions is extracted using a neural network for defining the confidence of

the new transactions. Recently, [15] used an automatic deep learning-based feature extraction to represent each transaction using 200 detailed features. The model was able to reduce 54% of the false positives predicted by the traditional model.

The binary classification applies a machine learning technique to predict whether a transaction is a fraud. The detection model is trained with labeled transactions. Dorronsoro et al. [5] applied a Multi-Layer Perceptron (MLP), which can perform real-time fraud detection. A system called CardWatch [6]; applied a feed-forward MLP of three-layers architecture. The proposed architecture can detect 85% of fraudulent transactions. The difficulty of the binary classification method is unbalanced of the labeled data between frauds and non-fraud transactions, which results in the classification of the non-fraud cases rather than the fraud cases. In the credit scoring domain, research has mainly focused on features that represent the behavior to be utilized in the prediction models. The performance for the minority class decreases significantly as the imbalance ratio increases [16], [17]. However, only a few works have addressed the design solutions for unbalanced credit data sets [18]. In addition, extracting good features that are able to separate fraud from legitimate transactions plays an important role.

Abnormal behavior can be considered an outlier. It as a transaction with characteristics that significantly deviated from the characteristics of inlier transactions as a fraud [14]. Applied a graph-based method for anomaly detection for determining financial frauds in money laundering on transactions [12]. Wu et al. Applied a convolutional algorithm that defined the boundary between normal and abnormal transactions, to detect frauds in an Automated Banking Machine (ABM) [11]. Applied a probabilistic model to compare the features of previously known transactions to find outliers [13]. Detected fraud transactions using the aggregated features to define the boundary of normal behavior for each account. Any transactions that are outside the boundary are considered fraudulent transactions [14]. There are many adoptions of deep neural networks in fraud detection using graph-based anomaly detection [18], [19]. Since transactions can be considered as sequential data, that are many methods that apply a Long Short-Term Memory (LSTM) network for fraud detection [20], [21]. One limitation of applying time series credit card detection is a lack of consistent patterns due to a limitation of proper data labeling of the huge dataset. Deep Anomaly Detection (DAD) was proposed [22] to track the customer's profile and usage behavior.

To measure the efficiency of the prediction model: the True Positive Rate (TPR), False-Positive Rate (FPR), and True Negative Rate (TNR) are considered. TPR measures the efficiency that correctly identifies fraudulent transactions. In contrast, TNR measures the efficiency of correctly identifying legitimate transactions as non-fraud. Lastly, FPR shows the proportion of legitimate transactions that are incorrectly identified as fraud. Due to an extremely low number of known fraud transactions, a prediction model can correctly predict one of the two classes. For example, many transactions are predicted as fraud, while they are non-fraud cases and overlook fraud cases, creating losses to the bank. In this work, a high number of TPR with a low FPR is preferred for prediction. The consequence of a false positive is the cost of investigating the transaction.

Two problems are being considered in this work. The first problem is unbalanced of the dataset. The previous research of our companion paper of [24] reported that the ratio of ATM fraud transactions from a bank in Thailand is less than 0.0002 of the total transactions. Sampling methods such as under and over-sampling were applied, but the accuracy is low. The second problem is the features representing customer behavior. Since many machine learning algorithms use numeric values for constructing models, non-numeric features were excluded.

In this work, we present a feature-based account-grouping method to cope with the unbalanced dataset problem. Also, the transformation functions are utilized for extracting numeric and non-numeric features. A descriptive analysis shows that the proposed features can separate suspicious transactions from normal transactions. Outlier detection methods, such as isolation forests and local outlier factors, are used to classify the frauds.

The experiments of the proposed methods are conducted on both supervised and unsupervised learning models. The results show that the proposed grouping method achieves higher TPR for both supervised and unsupervised methods. The FPR for the unsupervised methods is lower. Similarly, the proposed features achieve higher TPR.

The rest of this paper is organized as follows. Section 2 presents the result of a preliminary study and the previous works. Section 3 explains the data preprocessing. Section 4 presents the details of the feature extractions. Section 5 presents the feature-based sampling. The experimental result and the discussion are in Section 6. The conclusion is in Section 7.

II. Preliminary Study and Previous Works

A. Data Sampling

The popular sampling strategies for data consist of applying different forms of resampling to change the class distribution of the data. This can be done by either over-sampling the minority class or under-sampling the majority class until both classes are approximately equally represented [23].

Over-sampling is the simplest strategy that increases the amount of data in the minority class. It is a non-heuristic method that balances the class distribution through the random replication of positive examples. The drawback of the oversampling method is the original class distribution since it is artificially altered. In contrast, an under-sampling method may result in throwing out useful information about the majority class by randomly removing data. Despite its simplicity, it has empirically been shown to be one of the most effective resampling methods. However, the major problem with this technique is that potentially important data may be discarded in the prediction process. The study of [23] reported that an over-sampling method called SMOTE (Synthetic Minority Over-sampling TEchnique) proposed by [25] outperformed other sampling methods. It generates artificial examples from the minority class by interpolating the existing instances.

In this work, achieving a high true positive rate while keeping the number of false-positive lows is being focused on. The preliminary results for the no-sampling, under-sampling, and over-sampling methods are shown in Table I.

TABLE I
Prediction Result Using Different Sampling Methods

Sampling Method	Classifier	TPR	FPR	TNR
No Sampling	Neural Network	0.00%	0.00%	100.00%
	Random Forest	5.88%	0.00%	99.99%
	Isolation Forest	56.25%	2.47%	97.52%
	Overall	20.71%	0.82%	99.17%
Under-Sampling	Neural Network	90.63%	47.04%	52.95%
	Random Forest	56.25%	1.15%	98.84%
	Isolation Forest	0.00%	0.67%	99.32%
	Overall	48.96%	16.29%	83.71%
Over-Sampling	Neural Network	90.63%	47.14%	52.85%
	Random Forest	0.00%	0.14%	99.85%
	Isolation Forest	12.50%	4.72%	95.27%
	Overall	34.37%	17.34%	82.66%

Without sampling, the model only predicted the transactions as non-fraud. Therefore, the TNR of the three models was high which are 100.00%, 99.99%, and 97.52% for the neural network, random forest, and isolation forest, respectively. Comparing the over-sampling to the under-sampling, the under-sampling using random forest achieved the best result which that was 56.25% TPR and 1.15% FPR. Although the TPR of the random forest is lower than the neural network, the false positive rate shows that the random forest performed better.

The overall performance is computed from the average true positive, false positive, false negative, and true negative of the three models for each sampling technique. The overall result shows that applying the sampling technique either an under-sampling or over-sampling technique achieved a better true positive rate. From the preliminary result, we can conclude that utilizing a sampling technique can improve the performance of the prediction model, to achieve a high true-positive rate while keeping a low false-positive rate.

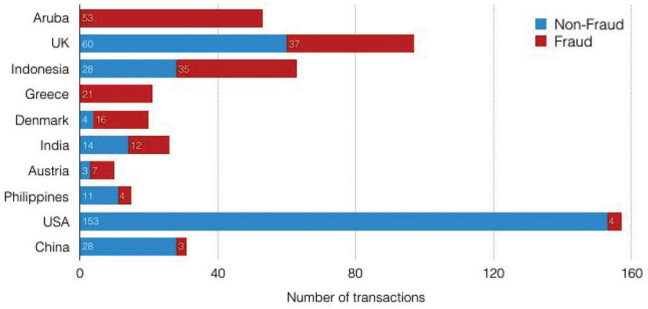
In our preliminary study in [24], there were 227 fraud transactions, where 35 took place locally in Thailand and 192 were from abroad. The ratio of frauds, when compared to the total number of transactions in one year was 0.0033% for local transactions and 18.69% for abroad transactions. The transactions that took place abroad have a higher chance of being fraud, as shown in Fig. 1. In this way, utilizing a location feature can be used to separate the dataset.

B. Outlier Detection Methods

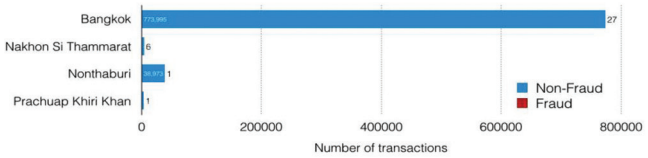
Many outlier detection methods were proposed to identify data points that do not conform to the normal characteristics by measuring distances or densities.

The Isolation Forest algorithm presented in [26] constructs a binary search tree to isolate outliers. It isolates data points in each tree level by selecting features and splitting values of the majority points to differentiate a common range of features from the uncommon. Fig. 2 illustrates the partition of a data point denoted by a rectangular point. If a data point is an outlier (as shown in Fig. 2 (b)), it is distinguishable in early partitioning. In other words, the number of partitions for the observation is less, as compared to the partition of Fig. 2 (a). Isolation Forest is an algorithm with low time and space complexity. It also has the ability to deal with high-dimensional data with irrelevant attributes. To get more accurate partitioning, Isolation Forest constructs multiple trees and averages the distances from the root node

to each leaf node (an isolated node). The number of samples and features that are picked for partitioning are random and different for each tree. In this way, the distance from the root node to each isolated node is different in each tree. The isolated nodes that are far from other nodes are isolated, to be nearby the root nodes of every tree.



(a) Number of legitimate and fraudulent transactions in different countries



(b) Number of legitimate and fraudulent transactions in Thailand

Fig. 1. Percentages of frauds and non-frauds in different locations

The Local Outlier Factor (LOF) presented in [27] is a density-based method that employs a *k*-nearest neighbors search. Each data point is scored by comparing the local density of the point, denoted by An in Fig. 3, with the local densities of its neighbors, denoted by *B*. The local density of *A* is the average reachable distance of *A* from its *k* neighbors. The reachable distance from *A* is defined by the maximum distance from *A* to *B* and *B* to the other neighbors such as *c*. In Fig. 3, the reachable distance from *A* is the distance from *A* to *B*.

The degree of being an outlier is the reachable distance from *A* to its nearest neighbors and from a neighbor *B* to its neighbors. For every neighbor *B* of *A*, the local density is the average reachable distance with each *k*-nearest neighbor's average reachable distance. If the average reachable distance is approximately equal to 1, the density of *A* is similar to its neighbor [28]. On the other hand, *A* is an inlier if the average reachable distance is less than 1. Otherwise, it is an outlier. In Fig. 3, the average reachable distance of *A* is greater than 1, therefore, *A* is considered as an outlier.

(a) Nominal Point (b) Anomaly Point

Fig. 2. (a) Partitions of a normal point. (b) Partitions of an anomaly point

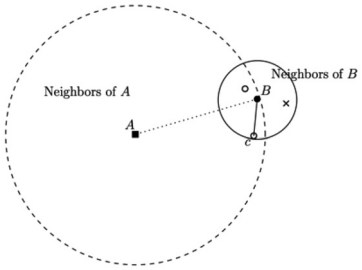


Fig. 3. An illustration of the local density of observation *A* to one of its k-nearest neighbors *B*

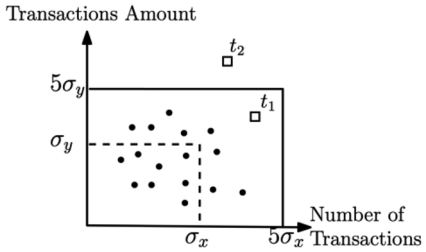


Fig. 4. Example of the normal behavior of an account within a fixed period

C. *Aggregated Features*

The normal pattern of transactions was extracted by aggregating numeric features extracted from the previous transaction during a fixed period to summarize the account behavior. In [4], the normal behavior was represented using the mean value of the numeric features such as the transaction amount and the number of transactions per day, which are

defined as μ_x and μ_y in Fig. 4. To identify fraud using the previous transactions, normal behavior was defined using μ_x and μ_y by defining a boundary from (0,0) to ($5\mu_x$,0) and from (0,0) to (0,$5\mu_y$). For an aggregate transaction, denoted by *t*, it is identified as a normal behavior if *t* lies within the solid boundary. See transaction t_1 in Fig. 4 for an illustration. The aggregated transactions of day t_2 are defined as fraud since it lies outside the solid boundary.

III. Preprocessing

A. ATM Transactions and Feature Selection

The ATM transactional data was obtained from a bank in Thailand. Due to the non-disclosure agreement, the name of the financial institution is omitted in this work. The ATM transactions were collected from 1/01/2017 to 31/12/2017. The number of transactions is 1,317,717.

For each transaction, there are 14 features, in which 2 features are numeric and the remaining 12 features are non-numeric values. The detail of each numeric feature is described below:

- Trans_Amnt: Amount of money in a transaction
- Fee_Amnt: Amount of fee, applied in a transaction

The detail of each non-numeric feature is described below:

- Card_No: Identification number of an ATM card
- Account_ID: Identification number of a bank account
- Trans_Date: Date of transaction
- Trans_Time: Time of transaction
- Service_Type: Service such as ATM, Account Setting, etc.
- Trans_Type: Transactions such as withdraw, deposit, etc.
- Trans_CD: Result of a transaction, e.g. success, failure, or rejection
- Trans_Reason: Reason for the response transaction
- Terminal_ID: Identification number of an ATM terminal
- Terminal_Bank: Owner of the ATM terminal
- Dest_Account_ID: Destination account number of the transaction
- Flag: Fraud label. For example, 'False' indicates a legitimate transaction.

In this work, behavior features of withdrawal transactions through an ATM are aggregated daily. Therefore, the non-numeric features, which are 'Service_Type', 'Trans_Time', 'Trans_Reason', and 'Dest_Account ID' are omitted and 10 raw features are used in this research.

B. Preprocessing

From the obtained data of 1,317,717 transactions, only withdrawal transactions, where the terminal country can be retrieved and the amount is greater than zero, are used. Therefore, 1,045,847 transactions are used, whereas 227 transactions of 43 accounts are frauds. The preprocessed features of each transaction that are used as input in this work are shown in Table II.

To represent the behavior of each account, features of 'date of the month' and 'day of the week' are derived from the 'Trans Date' feature. In addition, the location of the terminal ID is replaced with the corresponding coordinates.

In our preliminary study in [24], fraud transactions frequently took place abroad as shown in the descriptive analysis in Fig.1. Two new features, i.e. the 'distance' and the 'velocity' between the two consecutive transactions of each account are extracted. The intuition is the location of the two consecutive transactions should be varied to the interval between the transactions. Therefore, the latency of the two transactions in different countries or provinces should not be too close.

Let $\mathcal{T} = \{T_1, ..., T_n\}$ be a set of transactions of n accounts in the dataset. For each account i, let T_i be a set of k transactions, where $T_i = \{t_1,...,t_j,...,t_k\}$. Each transaction t_j is a set of features as shown in TABLE II. The transaction t_j is defined as $t_j = \{t^{acc}, t^{card}, t^{date}, t^{dom}, t^{dow}, t^{amt}, t^{cd}, t^{type}, t^{terID}, t^{terBank}, t^{loc}, t^{flag}\}$

Let $dist(t_{j-1},t_j)$ be the L_1 distance between the two consecutive transactions, i.e. and it is defined as follows:

$$dist(t_{j-1}, t_j) = |t_{j-q}^{loc} - t_j^{loc}| \quad (1)$$

where is the coordinates of the location where the transaction t_j took place. Let $velocity(t_{j-1}, t_j)$ be a latency between the two transactions. It is defined as the ratio of the distance to the number of days of the two consecutive transactions. It is defined as:

$$velocity(t_{j-1}, t_j) = \frac{dist(t_{j-1},t_j)}{days(t_{j-1}^{date},t_j^{date})} \quad (2)$$

where $dist(t_{j-1}, t_j)$ is the distance as defined in (1) and $days(t_{j-1}, t_j)$ is the number of days between the transactions t_{j-1} and t_j.

TABLE II
DESCRIPTION OF 14 FEATURES BEING USED IN THIS WORK

Features	Notation	Description
Account ID	t^{acc}	Account number of ATM card number.
Card No	t^{card}	Card number that was inserted.
Trans Date	t^{date}	Date of transaction.
Trans DOM	t^{dom}	Date of month.
Trans DOW	t^{dow}	Day of week.
Trans Amt	t^{amt}	Amount of money in the transaction.
Trans CD	t^{cd}	Result of the transaction.
Trans Type	t^{type}	Type of the transaction.
Terminal ID	t^{terID}	ID of ATM terminal.
Bank	$t^{terBank}$	Bank owner of the ATM terminal.
Location	tl^{oc}	Coordinate of a terminal's location.
Flag	t^{flag}	Flag indicating non-fraud (0) or fraud (1).
Distance	*dist*	Distance from the previous transaction.
Velocity	*velocity*	Latency of the previous transaction.

From the obtained 10 raw features, each transaction is transformed to find the distance and the velocity of the two consecutive transactions. Therefore, the features of the transaction t_j become: $t_j = \{t^{acc}, t^{card}, t^{date}, t^{dom}, t^{dow}, t^{amt}, t^{cd}, t^{type}, t^{terID}, t^{terBank}, t^{loc}, t^{flag}, dist, velocity\}$.

The details of the features being utilized in this work can be found in Table II.

IV. BEHAVIOR FEATURE EXTRACTION

Given a set of transactions $\mathcal{T} = \{T_1,...,T_n\}$ on n accounts, the transactions of each account T_i are preprocessed. An ATM transaction $t_j \in T_i$ of each account i has 14 features as explained in Section III, which are
$t_j = \{t^{acc}, t^{card}, t^{date}, t^{dom}, t^{dow}, t^{amt}, t^{cd}, t^{type}, t^{terID}, t^{terBank}, t^{loc}, t^{flag}, dist, velocity\}$

The 14 features show the details of each transaction. To represent whether the transaction is consistent with the normal behavior of the account or likely the result of a fraudster, four transformation functions, which are 'Normal Behavior', 'Deviation of Normal Behavior', 'Risk Value', and 'Probability', are derived. Each function is denoted by $g(f, i)$, $c(f, j, i)$, $r(\widehat{f_x})$, and $p(\widehat{f_x}, i)$, respectively, where f is a feature with a numeric value and $\widehat{f_x}$ is a non-numeric value in a transaction t_j of an account i. The summary of the transformed features that are used in this work is summarized in Table III. The details of each function are described in the following subsections.

TABLE III
SUMMARY OF THE DERIVED FEATURES OF A SET OF TRANSACTIONS T_I OF AN ACCOUNT I

Features	Description
$c(t^{amt}, i)$	Deviation from the average transaction amount
$c(t^{terBank}, i)$	Deviation from the average transaction amount of a bank
$p(t^{terBank}, i)$	Probability of using this bank
$r(t^{terBank})$	A risk score of this bank
$c(t^{terID}, i)$	Deviation from the average amount of a terminal
$p(t^{terID}, i)$	Probability of using this terminal
$r(t^{terID})$	A risk score of this terminal
$c(t^{loc}, i)$	Deviation from the average amount for a country
$p(t^{loc}, i)$	Probability of using this country
$r(t^{loc})$	A risk score of this country
$c(dist, i)$	Distance deviation from the average distance between consecutive transactions
$c(velocity, i)$	Velocity deviation from average velocity between consecutive transactions. (m/day)
$c(t^{dow}, i)$	A number of days that deviate from the average day of the week
$c(t^{dom}, i)$	A number of days that deviate from the average day of the month

A. Normal Behavior Value

To obtain normal behavior, only normal transactions are considered to characterize the normal behavior of each account. In other words, the transactions $t_j \in T_i$ in which $t^{flag} = 1$ are excluded from the normal behavior computation.

Given a set of normal transactions T_i of an account i, where $T_i = \{t_1,...,t_j,...,t_k\}$ and $t_j^{flag} = 0$ for $1 \le j \le k$, let $g(f,i)$ be an average value of a numeric feature f such as t_j^{amt} of account i, which is defined as follows:

$$g(f,i) = \frac{1}{|T_i|}\sum_{j \in T_i} f_i \quad (3)$$

where f_j is a numeric feature value of a transaction j.

The non-numeric features are transformed into numerical values by aggregating the transaction amount that occurred with the non-numeric values. For example, 10 transactions took place in Bangkok. The total amount of those transactions was 12,000. The non-numeric value of t^{loc} = Bangkok is 12000/10. Therefore, $g(\widehat{f_x})$ where x = Bangkok becomes 1200.

Let $g(\widehat{f_x}, i)$ be the average transaction amount of a non-numeric feature $\widehat{f_x}$ of value equal x such as 'Bangkok' for 'Terminal Location'. It can be defined using the following equation:

$$g(\widehat{f}, i) = \frac{1}{k}\sum_{j \in T_i(\hat{f})} t_j^{amt} \quad (4)$$

where k is the number of transactions such that $f \in T_i$ and $= x$.

In this paper, the normal behavior model of an account consists of aggregated features derived from t^{amt}, $dist$, $velocity$, t^{dom}, t^{dow}, t^{terID}, $tterBank$, and t^{loc}. Only t^{amt}, $dist$, and $velocity$, which are numeric values, are aggregated using $g(f, i)$. The remaining features apply $g(\widehat{f_x}, i)$.

B. Deviation from Normal Behavior Value

The deviation from the normal behavior value considers all the transactions of each account, including both legitimate and fraud. To represent the variation of each transaction t_j to the normal behavior of account i, a confidence value is computed using the normal behavior value to determine the deviation from the current transaction.

Let $c(f, j, i)$ be the confidence on the value of feature f for each transaction $t_j \in T_i$ in an account i. It is defined as follows:

$$c(f,j,i) = \frac{f - g(f,i)}{g(f,i)} \quad (5)$$

where $g(f, i)$ is the normal behavior value of the feature f as defined in (3). A non-numeric feature, such as the terminal location, is defined using $g(\widehat{f}, i)$, as defined in (4). The value of $\widehat{f}$ is the transaction amount occurring with the value of the non-numeric feature.

The confidence value $c(f,j,i)$ is computed for t^{amt}, $velocity$, $dist$, t^{dom}, t^{dow}, t^{terID}, $t^{terBank}$, and t^{loc}.

C. Risk Value

The risk value is the ratio of the total number of known fraudulent transactions to the total number of every transaction that occurred with the non-numeric feature $\widehat{f_x}$. It is determined from all transactions in the training dataset that are associated with all the non-numeric feature $\widehat{f_x}$ with a value x.

Let $r(\widehat{f_x})$ be the risk value for the feature $\widehat{f}$, with a value of x. Let $\mathcal{F}$ be the set of all fraudulent transactions and let $\mathcal{T}$ be a set of all transactions. The risk value is defined as follows:

$$r(\widehat{f_x}) = \frac{\sum_{l=1}^{|\mathcal{F}|} h(\mathcal{F},\widehat{f_x},t_l)}{\sum_{l=1}^{|\mathcal{T}|} h(\mathcal{T},\widehat{f_x},t_l)} \quad (6)$$

where, $h(\mathcal{F}, \widehat{f_x}, t_l)$ is the total number of known fraudulent transactions in $\mathcal{F}$ associated with the value of the feature $\widehat{f_x}$, and $h(\mathcal{T}, \widehat{f_x}, t_l)$ is the number of all transactions in $\mathcal{T}$ with the feature .

The number of fraud transactions $h(\mathcal{F}, \widehat{f_x}, t_l)$ is obtained using the following condition:

$$h(\mathcal{F}, \widehat{f_x}, t_l) = \begin{cases} 1 & \text{if } t_l^{\hat{f}} == x \\ 0 & \text{otherwise} \end{cases}$$

The value of $h(\mathcal{T}, \widehat{f_x}, t_l)$ can be obtained analogously using all transactions in $\mathcal{T}$.

For example, if $\widehat{f}$ = 'location' and x = 'Bangkok'. There are 10 transactions in in 'Bangkok', where 90 are legitimate and 10 are fraudulent transactions. The risk value of the of 'location' = 'Bangkok' is 0.1. The risk values are determined for non-numeric value $\widehat{f_x}$, of t^{terID}, $t^{terBank}$, and t^{loc}.

D. Probability

A probability feature shows how likely the account i will complete the transaction t_j regarding to the non-numeric feature $\widehat{f_x}$ of a value x.

Given a set of transactions T_i of the account i, a probability is a ratio of the number of transactions $t_j \in T_i$ that has the feature $\widehat{f} = x$ to the total number of transactions of T_i. Since fraud transaction is rare, the number of legitimate transactions that occurred with $\widehat{f_x}$ should be larger. In this way, the probability of a normal transaction is higher than fraud. The probability is defined as follows.

$$p(\widehat{f_x}, i) = \frac{\sum_{l=1}^{|T_i|} h(T_i,\widehat{f_x},t_l)}{|T_i|} \quad (7)$$

where,

$$h(T_i,\widehat{f_x}, t_l) = \begin{cases} 1 & \text{if } t_l^{\hat{f}} == x \\ 0 & \text{otherwise} \end{cases}$$

The probabilities are computed for non-numeric values, i.e., $\widehat{f_x}$ of t^{terID}, $t^{terBank}$, and t^{loc}.

V. Feature-Based Data Grouping

To achieve a high true positive rate while keeping the false positive rate as low as possible, accounts are separated into 'local-only' and 'has-abroad' groups based on the location feature, t^{loc}.

Given a set of transactions T_i of an account i and $T_i \in \mathcal{T}$, all transactions $t_j \in T_i$ is assigned to the 'has-abroad' group if there is a transaction t_j such that the location t_j^{loc} is not in Thailand. Otherwise, they are assigned to the 'local-only' group.

The number of transactions in the 'local-only' group is 1,025,216, where 16 transactions are frauds and the rest are legitimate transactions. For the 'has-abroad' group, there are 16,330 transactions, where 16 transactions are frauds and the rest are legitimate transactions.

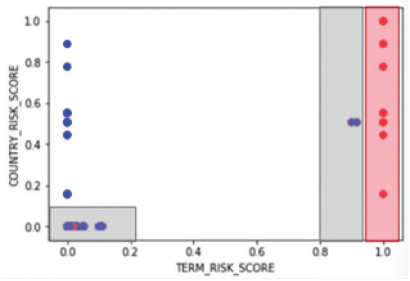
A. Analysis: Behavior Features and Data Grouping

To visualize the performance of the features in identifying fraud, a two-dimensional scatter plot of ATM transactions for terminal risk score and country risk score features is shown in Fig. 5(a). The frauds are in red and the normal transactions are in blue. From the graph, the transactions with a terminal risk score of 1 are frauds. However, there are some frauds whose feature values are close to the legitimate transactions that are depicted in Fig. 5(b). From the figure, it can be concluded that the separation between normal and fraudulent transactions is hard to be defined.

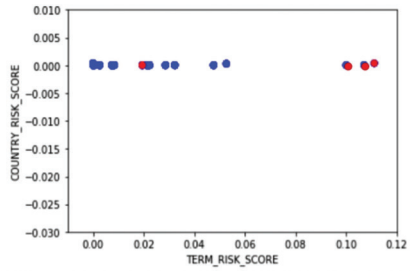
A plot of the country and the terminal risk scores of transactions in the 'has-abroad' group is shown in Fig. 6(a). Compared to the plot in Fig. 5(b), the fraud transactions in Fig. 6(a) can be separated. Using the other derived features such as the deviation from the average amount of a country and the terminal's risk score, the fraud transactions can also be separated as depicted in Fig. 6(b).

VI. Experiment

The ATM transactional data from 1/01/2017 to 31/12/2017 were obtained from a bank in Thailand. Due to the nondisclosure agreement, the name of the bank cannot be given. The objective of this work is to effectively detect fraud transactions while keeping a low number of false positives. The true positive rate (TPR), the false positive rate (FPR), and the true negative rate (TNR) are used to show the efficiency of the model.

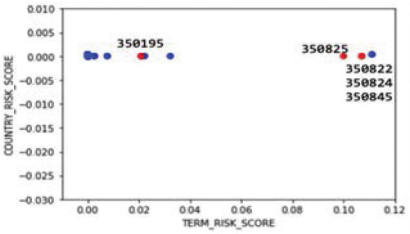


(a) Labeled ATM transactions of terminal risk score and country risk score.

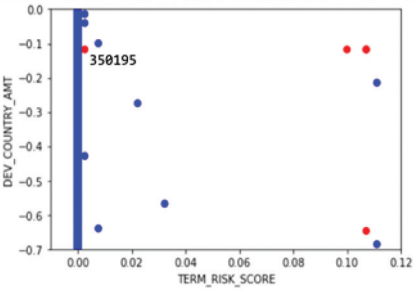


(b) The points in the lower-left shaded area of Fig. 5(a)

Fig. 5. Examples of fraud transactions (red) that are similar to legitimate transactions (blue).



(a) The transactions in the has-abroad group in the lower-left thevshaded area of Fig. 5(a)



(b) The transactions in the has-abroad account group with the terminal risk and deviation amount.

Fig. 6 Effects of the proposed risk score features for separating fraud transactions

Only withdrawal transactions, where the terminal country is known and the amount is greater than zero, are used. The total number of transactions in the experiment is 1,045,847. They are separated into the training and the testing sets. The training set contains the transactions from 01/01/2017 to 30/09/2017.

The transactions in the testing set of each account are transformed using the same functions. They are used to compare the result of the prediction model and the label assigned by the fraud investigators.

The testing set is the transactions from 01/10/2017 to 31/12/2017. The performance of the proposed features, the grouping using location-based features, and the classifiers including an outlier detection are being studied. Three patterns of the data grouping, i.e., 'no-grouping', 'local-only', and 'has-abroad' are used. In 'no-grouping', all the 1,045,847 transactions are used, where 790,984 (75.63%) transactions are for training and 254,863 (24.37%) are for testing. The 735,865 (71.77%) transactions in 'local-only' group are for developing a model and 289,351 (28.23%) are for testing. Among the testing transactions, 16 transactions were identified as frauds and confirmed by the bank. The 12,939 (79.23%) transactions in 'has-abroad' are for model development; 3,391 (20.77%) are for testing where 16 were identified as frauds by the bank.

Two outlier detection models: Local Outlier Factor (LOF) and Isolation Forest (IF) are employed. The two models are available in SciKit-Learn library [29]. Each model requires different parameters. LOF utilizes two parameters which are the number of neighbors and the contamination. The former parameter is the minimum number of k-nearest neighbors of each transaction required to classify the transaction as normal, while the latter is the estimated ratio of frauds in the population, which can be set between 0 and 0.5. IF requires three parameters. The first one is maximum number of samples for constructing each tree. The second is the maximum number of features to be used for each tree. Lastly, the contamination parameter is the ratio of fraud in the data set. In addition to the outlier detection methods, two binary classification models based on the artificial neural network (ANN) and Random Forest (RF) are employed. parameters settings for each experiment are in TABLE IV.

TABLE IV
PARAMETER SETTINGS FOR ATM FRAUD DETECTION

Model	Tuning Parameters		
	Local-Only	**Has-Aboard**	**No grouping**
LOF	n neighbors = 8, contamination = 0.1	n neighbors = 3, contamination = 0.4	n neighbors = 4, contamination = 0.5
IF	max samples = 1000, max features = 12, contamination = 0.045	max samples = 320, max features = 15, contamination = 0.0145	max samples = 1000, max features = 15, contamination = 0.0125
ANN	hidden layer size = 40, 20, 10, 10	hidden layer size = 40, 20, 10, 10	hidden layer size = 40, 20, 10, 10
RF	maximal depth=10, number of trees=10	maximal depth=10, number of trees=10	maximal depth=10, number of trees=10

TABLE V
CONFUSION MATRIX FOR FRAUD DETECTION

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	TP	FN
	Not Fraud	FP	TN

The classified result can be decomposed using a confusion matrix for fraud detection as shown in TABLE V. True Positive (TP) outcome occurs when a fraud transaction (positive) is correctly classified as a fraudulent transaction. A False Positive (FP) occurs when a non-fraudulent transaction (negative) is incorrectly identified as a fraudulent transaction. A False Negative (FN) occurs when a fraud transaction (positive) is incorrectly identified as a non-fraudulent transaction (negative). Lastly, a True Negative (TN) outcome occurs when a non-fraudulent transaction (negative) is correctly identified as a non-fraudulent transaction (negative).

The overall performance measures are true positive rate (TPR), false-positive rate (FPR), and accuracy. TPR is a ratio of the number of correctly classified as fraud transactions to the total number of transactions that are actually a fraud. It can be calculated as follow:

$$TPR = \frac{TP}{TP + FN}$$

FPR is a ratio between the number of not fraud transactions that are wrongly classified as fraud and the total number of actual 'not fraud' transactions. It can be calculated as follow:

$$FPR = \frac{FP}{FP + TN}$$

Accuracy is a ratio of the correctly classified transaction to the total number of transactions. It can be calculated as follow:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

A. Performance of the Sampling Methods

The results shown in Table VI are from the parameters settings of each model in Table IV. The binary classifier, the ANN, cannot detect any fraud transactions. In other words, every transaction is predicted as a legitimate transaction.

For the 'local-only' sampling method, the RF achieves 62.5% for detecting fraud, while the IF achieves a higher TPR of 75%. Comparing the false positive rate, IF is better than RF. However, the LOF obtains a lower FPR but it can correctly detect less than half of the fraud transactions.

For the 'has-abroad' sampling method, the IF can correctly detect all fraud transactions while the RF cannot detect any frauds. The IF can also achieve a low FPR of 8.23% that is lower than the LOF.

Table VII shows the confusion matrix of the IF classifier for detecting the fraud in the 'has-abroad' group. The result shows that it can correctly detect all 16 fraud transactions, while incorrectly detecting 278 normal transactions as frauds. The false-positive rate is 8.23%. The LOF detects 9 out of 16 fraud transactions but incorrectly detects 1,251 normal transactions as frauds.

Table VIII shows the confusion matrix of the IF for detecting the fraud in the 'local-only' group. It can detect 12 fraudulent transactions out of 16 transactions. However, 81,077 normal transactions are incorrectly detected as frauds, which results in 28.02% false-positive rate. The LOF model detects 7 out of 16 frauds and incorrectly detects 39,530 normal transactions.

TABLE VI
COMPARISON OF PERFORMANCE OF THE ATM FRAUD DETECTION MODEL OF 'LOCAL-ONLY' AND 'HAS-ABROAD'

Model	**Local-Only**		
	TPR	**FPR**	**TNR**
LOF	43.75%	13.66%	86.33%
IF	75.00%	28.02%	71.97%
ANN	0.00%	0.00%	100.00%
RF	62.50%	35.84%	64.15%
Model	**Has-Aboard**		
	TPR	**FPR**	**TNR**
LOF	56.25%	37.07%	62.93%
IF	100.00%	8.23%	91.76%
ANN	0.00%	0.00%	100.00%
RF	0.00%	1.30%	98.69%

TABLE VII
CONFUSION MATRIX OF ISOLATION FOREST (IF) FOR THE 'HAS-ABROAD' ACCOUNT GROUP

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	16	0
	Not Fraud	278	3,097

TABLE VIII
CONFUSION MATRIX OF ISOLATION FOREST FOR 'LOCAL-ONLY' ACCOUNT GROUP

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	12	4
	Not Fraud	81,077	208,258

TABLE IX
CONFUSION MATRIX OF ISOLATION FOREST FOR 'NO-GROUPING' ACCOUNT GROUP

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	24	8
	Not Fraud	32,172	260,538

Table IX shows the confusion matrix of the IF with 'no-grouping' transactions. The IF detects 24 out of 32 fraud transactions. However, it incorrectly detects 32,172 normal transactions as frauds, which results in a 10.99% false-positive rate.

The overall result of the proposed sampling method is compared with a method that does not apply any grouping. The average of the prediction result in the confusion matrix for true positive, false positive, false negative, and true negative of the 'local-only' and the 'has-abroad' groups of each model are computed. The result is shown as the average of two sampling methods in Table X.

The result shows that utilizing the proposed sampling method achieves better TPR for the IF of 87.50% and the false positive rate is 27.98%. The false-positive rate of the proposed grouping methods is higher than 'no-grouping', but the true positive rate is better. It can be concluded from the result that the proposed method is effective for the accounts that have transactions abroad.

TABLE X
COMPARISON OF PERFORMANCE OF THE AVERAGE OF THE 'LOCAL-ONLY' AND THE 'HAS-ABROAD' GROUPS WITH 'NO-GROUPING'

Model	**'Local-Only' And 'Has-Abroad'**		
	TPR	**FPR**	**TNR**
LOF	50.00%	13.93%	86.06%
IF	87.50%	27.98%	72.21%
ANN	0.00%	0.00%	100.00%
RF	31.25%	35.44%	64.55%
Model	**No Grouping**		
	TPR	**FPR**	**TNR**
LOF	31.25%	5.21%	94.78%
IF	75.00%	10.99%	89.00%
ANN	0.00%	0.00%	99.99%
RF	6.25%	5.31%	94.69%

TABLE XI
COMPARISON OF PERFORMANCE OF RAW FEATURES AND THE PROPOSED FEATURE EXTRACTION

Features	Classifier	TPR	FPR	TNR
Raw	ANN	0.00%	0.00%	100.00%
	RF	5.88%	0.00%	99.99%
	IF	56.25%	2.47%	97.52%
Proposed Feature	ANN	0.00%	0.00%	99.99%
	RF	6.25%	5.31%	94.69%
	IF	75.00%	10.99%	89.00%

B. Performance of Feature Extraction

The performance of the proposed feature extraction is compared using 'no-grouping'. Comparing 'no-sampling' with the raw feature is shown in Table XI. The proposed feature, applied with the IF, can detect 75.00% TPR and 10.99% FPR. Although the false-positive rate of the proposed feature is higher than using the raw feature, the performance for directly detecting fraud transactions correctly is much better.

C. Analysis

The experimental results show that the IF outperforms the other models. To identify an outlier, the IF looks through every tree and computes the average path length from the root to the isolation node. If the average path length of the observation is shorter than the average path length of a normal transaction, the observation is identified as an outlier. A fraud transaction as in Fig. 6 (a) is not located at the bottom of the tree. Apart from selecting the country and the terminal risk scores as the features, it selects the terminal amount deviation and the terminal risk scores features, as shown in Fig. 6 (a), in the other trees. Using these features, a fraud transaction is separated from normal transactions.

The LOF reports an outlier using the density of an observation and its *k*-nearest neighbors. From Fig. 6(b), the average density of the fraud transaction to its *k*-nearest neighbors is close to one. Therefore, it could not detect 350,195 transactions as fraud.

Comparing our behavior features to the previous work in credit card fraud [3], the proposed method requires transactions of one account owner for a longer period than the previous work that utilized only 7 days. This limitation is caused by the number of transactions of ATM being lower than the credit card one card owner may have many cards but only a few accounts are owned by the same customer.

VII. CONCLUSION

Three features: the deviation, the risk, and the probability features for both numeric and non-numeric values are presented. The proposed features can separate legitimate transactions from frauds. The accounts are separated into two groups depending on the location of the previous transactions, which are 'local-only' and 'has-abroad' groups. The experimental result shows that the 'has-abroad' group can improve the true positive rate of detecting fraud from 75% to 100% TPR when applying the isolation forest outlier detection method. Moreover, the false positive rate is 8.23%.

From the experimental result, the proposed features and the location-based grouping is performed well on the transaction that occurred abroad. Compare to the previous research on fraud detection, the proposed location-based grouping extracts the behavior features of the accounts in the same group, while the previous works use the behavior of every account or credit card. To improve the performance of the transactions that occurred locally, like in Thailand, finer details of the location should be mentioned such as a place with many tourists or an isolation location of the ATMs.

ACKNOWLEDGEMENTS

This work was supported by an SIIT Young Researcher Grant, under contract no. SIIT 2017-YRG-NK04. The authors would like to thank Prof. Dr. Thanaruk Theeramunkong, Dr. Thepchai Supnithi, and Assoc. Prof. Dr. Nobuhiko Sugino for their fruitful advice and comments.

REFERENCES

[1] Ben Knieff, "2016 Flobal Custumer Card Fraud: Where Card Fraud is Coming From," Aite Group LLC., Boston, USA, Jul. 2016.

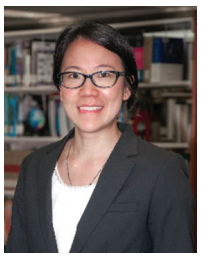
[2] T. Guo and G.-Y. Li, "Neural Data Mining for Credit Card Fraud Detection," in *Proc. 2008 International Conference on Machine Learning and Cybernetics*, 2008, pp. 3630-3634.

[3] C. Whitrow, D. J. Hand, P. Juszczak et al., "Transaction Aggregation as a Strategy for Credit Card Fraud Detection," *Data Mining and Knowledge Discovery*, vol. 18, no. 1, pp. 30-55, Feb. 2009.

[4] M. Krivko, "A Hybrid Model for Plastic Card Fraud Detection Systems," *Expert Systems with Applications*, vol. 37, no. 8, pp. 6070-6076, Aug. 2010.

[5] J. R. Dorronsoro, F. Ginel, C. Sgnchez et al., "Neural Fraud Detection in Credit Card Operations," *IEEE Transactions on Neural Networks*, vol. 8, no. 4, pp. 827-834, Jul. 1997.

[6] E. Aleskerov, B. Freisleben, and B. Rao, "Cardwatch: A Neural Network Based Database Mining System for Credit Card Fraud Detection," in *Proc. The IEEE/IAFE 1997 Computational Intelligence for Financial Engineering*, 1997, pp. 220-226.
[7] K. K. Sherly and R. Nedunchezhian, "Boat Adaptive Credit Card Fraud Detection System," in *Proc. 2010 IEEE International Conference on Computational Intelligence and Computing Research*, 2010, pp. 1-7.
[8] J. Gehrke, V. Ganti, R. Ramakrishnan et al., "Boat—Optimistic Decision Tree Construction," *SIGMOD Rec.*, vol. 28, no. 2, pp. 169-180, Jun. 1999.
[9] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data Mining Techniques for the Detection of Fraudulent Financial Statements," *Expert Systems with Applications*, vol. 32, no. 4, pp. 995-1003, May. 2007.
[10] E. Ngai, Y. Hu, Y. Wong et al., "The application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and An Academic Review of Literature," *Decision Support Systems*, vol. 50, no. 3, pp. 559-569, Feb. 2011.
[11] S. X. Wu and W. Banzhaf, "Combatting Financial Fraud: A Coevolutionary Anomaly Detection Approach," in *Proc. The 10th Annual Conference on Genetic and Evolutionary Computation*, SER. GECCO '08, 2008, pp. 1673-1680.
[12] D. Huang, D. Mu, L. Yang et al., "Codetect: Financial Fraud Detection with Anomaly Feature Detection," *IEEE Access*, vol. 6, pp. 19161-19174, Mar. 2018.
[13] K. Yamanishi, J. I. Takeuchi, G. Williams et al., "On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 275-300, May. 2004.
[14] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1-15:58, Jul. 2009.
[15] MIT. (2018, Oct. 2). *Reducing False Positives in Credit Card Fraud Detection.* [Online]. Available: https://www.sciencedaily.com/releases/2018/09/180920131513.htm
[16] D. J. Hand and V. Vinciotti, "Choosing K for Two-Class Nearest Neighbour Classifiers with Unbalanced Classes," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1555-1562, 2003.
[17] S. Bhattacharyya, S. Jha, K. Tharakunnel et al., "Data mining for Credit Card Fraud: A Comparative Study," *Decision Support Systems*, vol. 50, no. 3, pp. 602-613, Feb. 2011.
[18] S. Branka, J. B. K. Hofer-Schmitz, K. Nahrgang et al., "Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications," *Sensors*, vol. 21, no. 5, pp. 1594, Feb. 2021.
[19] T. Pourhabibi, K. L. Ongb, B. H. Kama et al., "Fraud Detection: A Systematic Literature Review of Graph-Based Anomaly Detection Approaches," *Decis. Support Syst*em, vol. 133, pp. 113303, Jun. 2020.
[20] W. Bao, J. Yue, and Y. Rao, "A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders and Long-Short Term Memory," *PLoS ONE*, vol. 12, pp. e0180944, Jul. 2017.
[21] A. Singh, "Anomaly Detection for Temporal Data Using Long Short-Term Memory (LSTM)," *IFAC-Papers Online* vol. 52, pp. 2408-2412, Jan. 2017.
[22] C. R. Chawla, "Deep Learning for Anomaly Detection: A Survey," *arXiv:1901.03407*, Jan. 2019.
[23] A. I. Marques, V. Garcıa, and J. S. Sanchez, "On the Suitability of˜ Resampling Techniques for the Class Imbalance Problem in Credit Scoring," *Journal of the Operational Research Society*, vol. 64, no. 7, pp. 1060-1070, Jul. 2013.
[24] R. Laimek and N. Kaothanthong, "Atm Fraud Detection Using Outlier Detection," in *Proc. IDEAL, 2018*, pp. 539-547.
[25] N. V. Chawla, K. W. Bowyer, L. O. Hall et al., "Smote: Synthetic Minority Over-Sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, Jun. 2002.
[26] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-Based Anomaly Detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 3:1-3:39, Mar. 2012.
[27] M. M. Breunig, H. P. Kriegel, R. T. Ng et al., "Lof: Identifying Density-Based Local Outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93-104, May. 2000.
[28] M. Elahi, K. Li, W. Nisar, X. Lv et al., "Detection of Local Outlier Over Dynamic Data Streams Using Efficient Partitioning Method," in *Proc. 2009 WRI World Congress on Computer Science and Information Engineering*, 2009, pp. 76-81.



Natsuda Kaothanthong received a Ph.D. in Information Science from Graduate School of Information Sciences, Tohoku University. She is an Assistant Professor in School of Management Technology at Sirindhorn International Institute of Technology, Thammasat University. Her research interests are machine learning, artificial intelligence, image processing, and medical images processing.

Roongtawan Laimek received a master degree in engineering from Thailand Advanced Institute of Science and Technology and Tokyo Institute of Technology.