SME User Classification from Click Feedback on a Mobile Banking Apps

Suchat Tungjitnob[1], Kitsuchart Pasupa[1(✉)][0000−0001−8359−9888], Ek Thamwiwatthana[2], and Boontawee Suntisrivaraporn[2]

[1] Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
62606062@kmitl.ac.th, kitsuchart@it.kmitl.ac.th
[2] Data Analytics, Chief Data Office,
Siam Commercial Bank, Bangkok 10900, Thailand
{ek.thamwiwatthana, boontawee.suntisrivaraporn}@scb.co.th

Abstract. Customer segmentation is an essential process that leads a bank to gain more insight and better understand their customers. In the past, this process requires analyses of data, both customer demographic and offline financial transactions. However, from the advancement of mobile technology, mobile banking has become more accessible than before. With over 10 million digital users, *SCB easy app* by Siam Commercial Bank receives an enormous volume of transactions each day. In this work, we propose a method to classify mobile user's click behaviour into two groups, *i.e.* 'SME-like' and 'Non-SME-like' users. Thus, the bank can easily identify the customers and offer them the right products. We convert a user's click log into an image that aims to capture temporal information. The image representation reduces the need for feature engineering. Employing ResNet-18 with our image data can achieve 71.69% average accuracy. Clearly, the proposed method outperforms the conventional machine learning technique with hand-crafted features that can achieve 61.70% average accuracy. Also, we discover a hidden insight behind 'SME-like' and 'Non-SME-like' user's click behaviour from these images. Our proposed method can lead to a better understanding of mobile banking user behaviour and a novel way of developing a customer segmentation classifier.

Keywords: Customer Segmentation · Click Feedback · Encoding Click Feedback as image · Mobile Banking Behaviour.

1 Introduction

Mobile banking has become widely: with rapid improvement of smartphone technology, it is more accessible than before. Through mobile banking apps, bank customers can immediately make banking transactions, anywhere and anytime. These apps also enable banks to reach their customers more efficiently and reduce costs for expanding additional branches. Thus, the banking industry has rapidly shifted from traditional banking services to digital platforms.

Customer segmentation splits customers into distinct groups, based on similar characters and behaviours, for example, job, salary, age, etc [10]. Currently, big data technology and data science enables us to obtain many insights from the data. This helps banks to market products appropriately. Existing work has demonstrated that user feedback leads to a better understanding of user behaviour [11–13], but it requires more effort to process the data. Obviously, user feedback benefits the bank industry, Sunhem and Pasupa have used a smartphone's front-facing camera to track eye movements for user experience and user interface survey research for mobile banking app [14]. In addition to feedback from eye movements, user click are also relevant and easy to collect in a system log. With more than 250 million transactions each month, *SCB easy app* by Siam Commercial Bank allows collection of a large database, giving the bank many opportunities to understand customer lifestyle and their preferences.

Currently, SCB provides many products for Small and Medium-sized Enterprises (SME), which are registered businesses that maintain revenues, assets or have several employees below a certain threshold. SCB encourages its customers to use these products and reach out to the companies. However, many customers are natural persons, who not registration as a company. Here, we describe them as 'SME-like'. The bank wants to encourage the 'SME-like' organisations to use SME products as well. It is challenging for the bank to identify this group of customers, unless the customer notifies it. Mobile banking app is currently available only for personal account but not for business account. In this work, we showed how to classify *SCB easy app* users into two groups, *i.e.* 'SME-like' and 'Non-SME-like', based on behaviour on the apps, *e.g.* user clicks. We examined user click logs for some anonymous *SCB easy app* users, that were labelled as 'SME-like' and 'Non-SME-like'. So that the model would enable identification of unlabelled users and allow the bank to offer the right products or services to the right customers. The contributions of our work are:

1. We described a set of hand-crafted features extracted from *SCB easy app* click log data to distinguish between 'SME-like' and 'Non-SME-like' users. These features were evaluated with the existing, Extreme Gradient Boosting algorithm [1].
2. Click log data were visualised in a way that captured temporal information for click log data. These images were used to train a Convolutional Neural Network (CNN) model. This representation performed better than hand-crafted features.

2 Related Work

Now we can access big data and traditional methods have limitations facing very large amounts of data [4]. For this reason, machine learning techniques have been applied to the customer segmentation problem, due to their effectiveness and robustness. Numerous machine learning techniques, including classification, *i.e.* decision tree [3], and clustering, *i.e.* k-means clustering [9] were used to segment customers in various industries. Kim *et al.* combined multiple classification methods to predict customer purchase propensity on an e-commerce website [7]. In the telecommunication industry, Dullaghan *et al.* analysed mobile customer behaviour and use by decision trees and Naïve Bayes algorithm [3].

Machine learning techniques have been used to segment customers in the banking industry as well. Li *et al.* segmented customer groups with credit card usage data by k-means clustering and then compared classification algorithms to train a forecasting model to separate unseen data according to those clusters [8]. Mihova *et al.* used k-means clustering to determine customer groups, based on their credit history [9].

Many works also showed that temporal data such as website click feedback can assist customer segmentation. Bock *et al.* trained a random forest classifier, with website visitor click feedback patterns. The classifier was able to form a demographic profile of a website visitor *i.e.* gender, age and occupation, to support an advertisement team [2].

Our click log data included time intervals, whereas most existing banking industry work did not consider this click data. Moreover, our click data came from a mobile banking application, whereas most previous work using temporal data came from website click data. Involving a usage time makes it possible to visualise these data in time and assist the management team to plan strategy.

3 Methods

3.1 Data Collection

The click log data come from a database of *SCB easy app*. They were flagged using data from user accounts as 'SME-like' and 'Non-SME-like'. The data was collected between 1–30 September, 2019. The raw log data labeled click events with a type and a timestamp. We collected data from 25,868 anonymous users, including 12,621 'SME-like' users and 13,247 'Non-SME-like' users, for 72,874,972 clicks. There were 535 unique events, which can divided into interface events and banking function events. Moreover, we wanted to select only events, that were essential to our task. So we created a matrix of our data by counting the number of usage times of each event of each user in one month, leading to a $25,868 \times 535$ matrix. This matrix represented the click event frequency in one month for each user. We used a random forest algorithm, using this matrix, to calculate an importance score for each event. We rank each event importance score and used an elbow method to determine a number of events that should be kept, based on its importance score: this led us to 56 events, including 21 banking function events and 35 interface events. We needed a small number of quality features to represent our target user and interface events did not represent mobile banking usage behaviour. Thus, we kept only 21 banking events and eliminated interface events. Then, we grouped the remaining 21 banking events into seven group of events, which we labeled 'primary events'. Events, that contained a keyword from one of the seven primary-events, were labeled as sub-events, *i.e.* transfer is a primary-event and transfer_slip was its sub-event.

We also included additional sub-events, eliminated by the elbow method, to be part of a primary event. Finally, our click log data contained 17,133,290 clicks, separated into seven primary events and 57 sub events, see Figure 1.

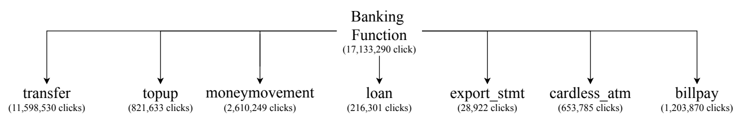

Fig. 1: Banking function primary events

3.2 Hand-crafted Feature Extraction

In this step, we extracted a feature from a click log data to distinguish between an 'SME-like' and 'Non-SME-like' *SCB easy app* users. We found three groups of hand-crafted feature methods presented in Figure 2.

1. Number of clicks for each primary event $\{F_1, F_2, \ldots, F_7\} \in \mathcal{F}_f$: We count a frequency for each primary event assuming that $\mathcal{F}_f$ should have a unique attribute, which can distinguish between 'SME-like' and 'Non-SME-like' users.
2. Number of clicks for each primary event within a period of time $\{F_8, F_9, \ldots, F_{28}\} \in \mathcal{F}_t$: We crafted a feature to included period of time due to the timestamp in our click log data. We counted each primary event frequency, using the timestamps, into three periods: 08:00 to 17:00, 17:00 to 24:00 and 24:00 to 08:00. This eight hour gap come from a standard working hour in Thailand, assuming that mobile banking application usage should be related to user working hour periods.
3. Combination of $\mathcal{F}_f$ and $\mathcal{F}_t$ ($\mathcal{F}_c$): We concatenated $\mathcal{F}_f$ and $\mathcal{F}_t$, assuming that a model would perform better when trained with more features.

$\mathcal{F}_f$		$\mathcal{F}_t$	
F_1	billpay	$F_8 - F_{10}$	billpay with time stamp
F_2	cardless atm	$F_{11} - F_{13}$	cardless atm with time stamp
F_3	export stmt	$F_{14} - F_{16}$	export stmt with time stamp
F_4	loan	$F_{17} - F_{19}$	loan with time stamp
F_5	money movement	$F_{20} - F_{22}$	money movement with time stamp
F_6	topup	$F_{23} - F_{25}$	topup with time stamp
F_7	transfer	$F_{26} - F_{28}$	transfer with time stamp

Fig. 2: Features extracted from click log data

3.3 Visualisation of the log

Since the click log contained time stamps, when we extracted a feature with the hand-crafted method described above, we had to select the number of time intervals used to extract features. Since, there are many ways of designing features with these time intervals, we converted the click logs into an image, which could capture the attribute of those time stamps and enable us to visualise and identify the use behaviour trends found in the logs.

Based on 24 hour days and 60 minute hours, we created 'images' from each primary event with 24×60 pixels. These images were very sparse, so we rescaled from 60 (1 min per pixel) to 30 (2 min per pixel) leading to 24×30 pixel images. The steps for representing the log as an image are shown in Figure 3.

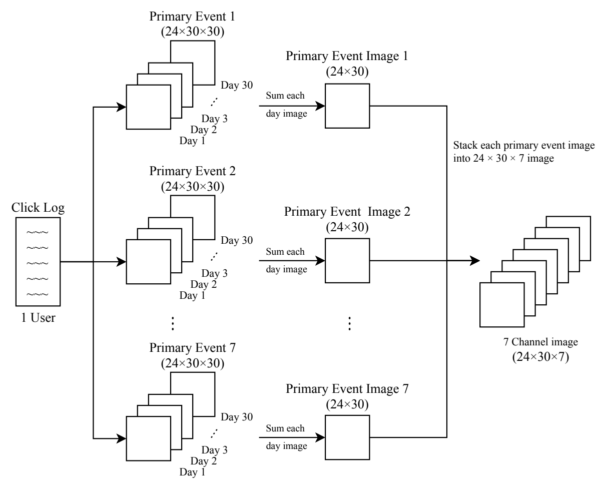

Fig. 3: Steps for encoding click log data in a seven channel image

The encoding led to seven 24×30 images, which represent use behaviour for each primary event of a single user. We stacked these seven images into one image, with seven channels—$(24 \times 30 \times 7)$ and refer them as $\mathcal{I}_{7c}$. Each channel of $\mathcal{I}_{7c}$ are describe in Figure 4. We visualised $\mathcal{I}_{7c}$ by extracting each channel into seven grey scale images. Each image represented a primary event behaviour of that user. When viewing the image, we can see a different use pattern for each user. For example in Figure 4, an 'SME-like' user did not use a cardless ATM function, so this image was all black, but a 'Non-SME-like' user used this event frequently, so this image tells us that this 'Non-SME-like' user frequently withdrew money, with the *SCB easy app* cardless ATM function.

Typical computer monitors can only basically output a three channel (RGB) image, but for $\mathcal{I}_{7c}$, we need to visualise each channel as a grey scale image.

Thus, $\mathcal{I}_{7c}$ was still difficult for a human to visualise and we devised a method that converts $\mathcal{I}_{7c}$ into RGB images, which are easier to visualise.

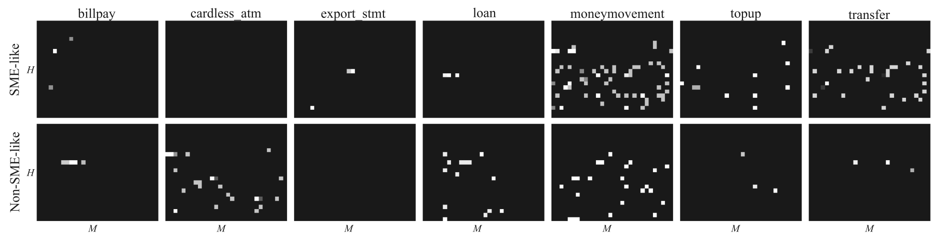

Fig. 4: Example of $\mathcal{I}_{7c}$: each row represents a primary event. Each column represents $\mathcal{I}_{7c}$ of one user example. H denote a 24 hour (1 hour per pixel). M denote a 30 intervals (2 minute per pixel).

We selected only the top three frequency channels ('transfer' (11,598,530 clicks), 'moneymovement' (2,610,249 clicks) and 'billpays' (1,203,870 clicks)) from $\mathcal{I}_{7c}$ to be represented as an RGB image (see Figure 1). This led to RGB images ($24 \times 30 \times 3$), defined as $\mathcal{I}_{rgb}$. Selection steps for the top three frequency channels, that convert $\mathcal{I}_{7c}$ into $\mathcal{I}_{rgb}$, are shown in Figure 3. Instead of forcing us to distinguish seven grey levels from $\mathcal{I}_{7c}$, $\mathcal{I}_{rgb}$ enabled us to easily visualise a behaviour from one RGB image.

3.4 Models

XGBoost is an optimised version of tree based gradient boosting [5], designed to be efficient, flexible and more regularised to control over-fitting. XGBoost also trains faster with better performance than other gradient boosting implementations [1]. We used XGBoost to evaluate the performance of $\mathcal{F}_f$, $\mathcal{F}_t$ and $\mathcal{F}_c$.

ResNet is a deep CNN architecture which was designed to eliminate the vanishing gradient problem. He *et al.* introduced a concept of 'skip connection', by adding an input value to the output after convolution block [6]. This enabled a network to stack more and deeper layers, without affecting training performance. We used an 18-layer version of ResNet, called ResNet-18. He *et al.*'s default settings [6] were used in an experiment on $\mathcal{I}_{rgb}$. For $\mathcal{I}_{7c}$, we modified ResNet-18 input layer settings to input a seven channel image, instead of the original three channel one.

4 Experiment Setting

Initially, we screened our data and found that many users were not very active and tended to provide noisy data, *i.e.* these users rarely used the *SCB easy* *app*. They not contribute useful information and their click log behaviours were difficult to distinguish.

Therefore, we only want to evaluate active users due to the reason that an active user is groups of users that are familiar to *SCB easy app* which can show a more meaningful usage behaviour. The question arose that where should we define this active threshold to discarded non-active users. We decided to sort our user usage frequency and discarded a user with lower than 50%, 55%, 60%, 65%, 70% and 75% percentile. The number of remaining users after discarding non-active users in each threshold are:

- 6,303 SME-like and 6,619 Non-SME-like for 50% threshold.
- 5,674 SME-like and 5,949 Non-SME-like for 55% threshold.
- 5,043 SME-like and 5,299 Non-SME-like for 60% threshold.
- 4,414 SME-like and 4,636 Non-SME-like for 65% threshold.
- 3,784 SME-like and 3,972 Non-SME-like for 70% threshold.
- 3,155 SME-like and 3,308 Non-SME-like for 75% threshold.

Finally, we conducted an experiment to evaluate our 5 set of inputs on SME-like classification task with 6 different discarded threshold.

Every input ($\mathcal{F}_f$, $\mathcal{F}_t$, $\mathcal{F}_c$, $\mathcal{I}_{7c}$ and $\mathcal{I}_{rgb}$) was first split into training and test sets with an 80:20 ratio. Then 20% of a training set was extracted as a validation set. The validation set was used for selecting a model with the lowest validation loss. We then evaluated the selected model on an unseen test set. All inputs were split with the same random seed to verify that every test set came from the same users, for the total of five random seeds.

For ResNet-18, the model settings were: batch size = 64; learning rate = 0.001 with the Adam optimisation algorithm; maximum number of epochs = 50.

5 Results & Discussion

Each input was trained with a different discarded threshold and the accuracy was evaluated on an unseen test set—see Table 1.

Table 1: 'SME-like' classification accuracy (as %)

Discarded threshold	Features				
	$\mathcal{F}_f$	$\mathcal{F}_t$	$\mathcal{F}_c$	$\mathcal{I}_{7c}$	$\mathcal{I}_{rgb}$
50	61.10±0.79	61.10±0.77	61.31±0.89	69.43±0.60	68.81±0.99
55	60.92±0.95	61.10±0.93	61.49±0.56	70.89±0.55	68.59±2.17
60	**62.29±0.80**	**62.34±0.63**	**62.37±1.08**	71.96±1.61	71.50±1.39
65	61.29±0.80	62.32±0.91	62.31±0.53	72.66±0.93	72.54±2.15
70	60.99±0.31	61.69±0.49	62.00±0.58	73.41±0.71	72.70±1.28
75	61.48±1.07	62.19±1.19	62.26±1.39	**73.46±2.43**	**74.31±1.35**
Average	61.35±0.79	61.79±0.82	61.95±0.84	71.97±1.14	71.41±1.56

From Table 1, considering only hand-crafted features, it can be seen that $\mathcal{F}_c$ achieved the best performance at 61.95% average accuracy, followed by $\mathcal{F}_t$ at 61.79% and $\mathcal{F}_f$ at 61.35%. In addition, when considering each discarded threshold, $\mathcal{F}_c$ also performed better than $\mathcal{F}_t$ and $\mathcal{F}_f$. This showed that $\mathcal{F}_t$, to which we added click frequency with time intervals, delivered a better performing model than $\mathcal{F}_f$. Lastly, $\mathcal{F}_c$ showed that our classifier gained more benefit from combining the hand-crafted features, which provided the classifier with extra information.

From ResNet-18, trained with image inputs in Table 1, $\mathcal{I}_{7c}$ achieved the best performance at 71.97% average accuracy, followed by $\mathcal{I}_{rgb}$ at 71.41%. $\mathcal{I}_{7c}$ performed better than $\mathcal{I}_{rgb}$ at every threshold, except the 75 percentile. In general, $\mathcal{I}_{7c}$ gained more performance from its seven channel information compared to $\mathcal{I}_{rgb}$ with only three channels. However, $\mathcal{I}_{rgb}$ was more suitable for visualisation and these two performed similarly, differing by only $\sim 0.56\%$ in average accuracy. Lastly, at each discarded threshold, we can see that by discarding more inactive users, we gained more performance in $\mathcal{I}_{7c}$ and $\mathcal{I}_{rgb}$, *i.e.* inactive users affected the model performance and need to be filtered out.

Overall, converting the click log data into $\mathcal{I}_{7c}$ and $\mathcal{I}_{rgb}$ improved the model performance: it outperformed $\mathcal{F}_f, \mathcal{F}_t$ and $\mathcal{F}_c$ at every threshold. The image captured better time-related features and reduced the need for hand-crafted features. Further, the image representation made visualisation of these logs much easier and helped display hidden patterns that lay in our click logs.

6 Conclusions

The click log data from a mobile banking application can lead to many opportunities for a bank to gain more knowledge of their customer. Converting these logs to images reduced the need for hand-crafted features and make it easier for understand user behaviour. Therefore, we proposed five sets of inputs, consisting of hand-crafted features ($\mathcal{F}_f, \mathcal{F}_t$ and $\mathcal{F}_c$) and images ($\mathcal{I}_{7c}$ and $\mathcal{I}_{rgb}$) to evaluate 'SME-like'classification. We found that using the images outperformed hand-crafted features, because it captured more time-related features. $\mathcal{I}_{rgb}$ performed better, when it come to visualisation and were suitable for helping the management team identify trends behind user click behaviour. We also investigated and provided some insights for the bank, which can help them able to reach more customer and increase their sales. Planned future work includes a multi-view approach generating another view of an image, for example, an image representation of behaviour in each business quarter and using this new image view, combined with the old one, to boost the performance of our 'SME-like' classifier.

References

1. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), San Francisco, CA, USA. pp. 785–794 (2016). https://doi.org/10.1145/2939672.2939785

2. De Bock, K., Van den Poel, D.: Predicting website audience demographics forweb advertising targeting using multi-website clickstream data. Fundamenta Informaticae **98**(1), 49–70 (2010). https://doi.org/10.5555/1803672.1803677
3. Dullaghan, C., Rozaki, E.: Integration of machine learning techniques to evaluate dynamic customer segmentation analysis for mobile customers. International Journal of Data Mining & Knowledge Management Process **7**, 13–24 (2017). https://doi.org/10.5121/ijdkp.2017.7102
4. Florez, R., Ramon, J.: Marketing segmentation through machine learning models: An approach based on customer relationship management and customer profitability accounting. Social Science Computer Review **27**, 96–117 (2008). https://doi.org/10.1177/0894439308321592
5. Friedman, J.: Greedy function approximation: A gradient boosting machine. Annals of Statistics **29**, 1189–1232 (2001). https://doi.org/10.2307/2699986
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
7. Kim, E., Kim, W., Lee, Y.: Combination of multiple classifiers for customer's purchase behavior prediction. Decision Support Systems **34**, 167–175 (2003). https://doi.org/10.1016/S0167-9236(02)00079-9
8. Li, W., Wu, X., Sun, Y., Zhang, Q.: Credit card customer segmentation and target marketing based on data mining. In: Proceedings of the International Conference on Computational Intelligence and Security (CIS 2010), Nanning, China. pp. 73–76 (2011). https://doi.org/10.1109/CIS.2010.23
9. Mihova, V., Pavlov, V.: A customer segmentation approach in commercial banks. In: AIP Conference Proceedings. vol. 2025, p. 030003 (2018). https://doi.org/10.1063/1.5064881
10. Ngai, E., Xiu, L., Chau, D.: Application of data mining techniques in customer relationship management: A literature review and classification. Expert Systems with Applications **36**, 2592–2602 (2009). https://doi.org/10.1016/j.eswa.2008.02.021
11. Pasupa, K., Chatkamjuncharoen, P., Wuttilertdeshar, C., Sugimoto, M.: Using image features and eye tracking device to predict human emotions toward abstract images. In: Proceeding of the 7th Pacific Rim Symposium on Image and Video Technology (PSIVT 2015), Auckland, New Zealand. Lecture Notes in Computer Science, vol. 9431, pp. 419–430 (2016). https://doi.org/10.1007/978-3-319-29451-3_34
12. Pasupa, K., Sunhem, W., Loo, C.K., Kuroki, Y.: Can eye movement information improve prediction performance of human emotional response to images? In: Proceeding of the 23nd International Conference on Neural Information Processing (ICONIP 2017), Guangzhou, China. Lecture Notes in Computer Science, vol. 10637, pp. 830–838 (2017). https://doi.org/10.1007/978-3-319-70093-9_88
13. Pasupa, K., Szedmak, S.: Utilising kronnecker decomposition and tensor-based multi-view learning to predict where people are looking in images. Neurocomputing **248**, 80–93 (2017). https://doi.org/10.1016/j.neucom.2016.11.074
14. Sunhem, W., Pasupa, K.: A scenario-based analysis of front-facing camera eye tracker for ux-ui survey on mobile banking app. In: Proceedings of the 12th International Conference on Knowledge and Smart Technology (KST 2020), Pattaya, Thailand. pp. 80–85 (2020)