A NEW FRAMEWORK OF FEATURE ENGINEERING FOR MACHINE LEARNING IN FINANCIAL FRAUD DETECTION

Chie Ikeda, Karim Ouazzane and Qicheng Yu

School of Computing and Digital Media,
London Metropolitan University, London, UK

ABSTRACT

Financial fraud activities have soared despite the advancement of fraud detection models empowered by machine learning (ML). To address this issue, we propose a new framework of feature engineering for ML models. The framework consists of feature creation that combines feature aggregation and feature transformation, and feature selection that accommodates a variety of ML algorithms. To illustrate the effectiveness of the framework, we conduct an experiment using an actual financial transaction dataset and show that the framework significantly improves the performance of ML fraud detection models. Specifically, all the ML models complemented by a feature set generated from our framework surpass the same models without such a feature set by nearly 40% on the F1-measure and 20% on the Area Under the Curve (AUC) value.

KEYWORDS

Financial Fraud Detection, Feature Engineering, Feature Creation, Feature Selection, Machine Learning

1. INTRODUCTION

Online banking services has expanded rapidly, and in tandem, fraudulent activities via the internet and credit cards have increased substantially. According to Financial Fraud Action UK in 2020, the financial fraud losses registered a record high of £824.8 million in 2019 [1]. Payment card and remote banking account for 60% of the whole fraud losses. Evidently, the fraud detection system (FDS), used by many financial institutions, has not caught up with the advancement in fraud schemes. To address constant changes in fraud schemes, the FDS has incorporated machine learning (ML), but it is still challenging to reveal new fraudulent patterns by applying ML to raw data only.

The recent studies in financial fraud detection have further adopted feature engineering, which is an essential work in data preparation for ML. Feature engineering involves two main progresses: feature creation in which feature candidates are created from original data, and feature selection in which features are selected among the candidates as an input for ML.

Broadly, feature creation is classified into two types: feature transformation and feature aggregation. Feature transformation creates features by transforming original data using some functions, which typically adopt mathematical or statistical functions. The recent example in the field of financial fraud detection includes Bahnsenet al [2] who use the statistical function of the von Mises distribution to transform interval time between the last transaction and the latest

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 205-220, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101517

transaction by each individual customer. Feature transformation is also useful to convert values in categorical features into numerical values because ML algorithms unable to directly deal with categorical features. For instance, Dummy variables can represent a single class from a categorical feature by a set of binaries with the exact same information.

Feature aggregation creates features by aggregating some patterns observed from original data. Feature aggregation combines various features from multiple tables into a new summary form, e. g., average amount of transaction by each individual customer, and number of accesses to an online banking account per month. For example, Yesilkanat et al. [3] and Y.Xie et al [20] use feature aggregation to express a sequential pattern of transactions and create new features by combining original data such as the place (such as an ATM location), the amount, and the time of transaction.

Feature selection – another progress in feature engineering - selects relevant features from the candidates created in feature creation for ML algorithms. By doing so, it addresses two issues: effectiveness and compatibility. It selects effective features that improve ML model predictions. It also makes features readily useable for a different type of ML algorithms.

In financial fraud detection, a variety of ML algorithms have been used. They include support vector machine (SVM), random forests (RF), logistic regression (LR), K-means, local outlier factor (LOF), neural networks (NN). These ML algorithms are broadly classified into two types: supervised learning and unsupervised learning. Supervised learning uses historical transaction records including a fraud flag and learns the different patterns between fraud and non-fraud data, while unsupervised learning deals with big data and observes latent patterns without learning fraud flags from past data. Unsupervised learning has more potential to reveal underlying fraud patterns than supervised learning by multiplying data without training. Lee et al. [4] use a feature selection process for unsupervised learning for credit card fraud detection and show that a detection accuracy of the unsupervised learning model with selected features is better than that of the same model but without feature selection. Varmedja et al. [5] use a feature selection process for supervised learning models such as Naïve Bayes (NB) and LR, and show the effectiveness with selected features.

Despite these progresses in the field of financial fraud detection, in the process of feature creation, most studies use either feature aggregation or feature selection separately.

Even if one type of feature creation is used, few studies use feature selection before putting features into ML models. Conversely, even if feature selectin is used, few studies use feature creation before selection features; most of the studies select variables from original data.

Against the background, in this paper, we propose a new framework of feature engineering for ML in financial fraud detection. Specifically, our framework consists of feature creation process and feature selection process jointly. In feature creation process, both techniques of feature aggregation and feature transformation are used to create feature candidates, which could improve an accuracy of ML models. Subsequently, feature selection process evaluates the candidate features in terms of classification report and the Area Under the Curve (AUC). Features are then selected based on the evaluation and are used as an input for appropriate ML algorithms.

The salient aspect of this framework is three-fold. First and most importantly, the combination of creation process and selection processes: use of feature aggregation and feature transformation jointly to create important feature candidates, and selection from the feature candidates based on evaluation by specific ML models. Second, in feature selection process, we consider compatibility between features and individual ML algorithm and built the framework that can accommodate any ML fraud detection models, which does not rely on a certain specific ML model. Third, few studies of feature engineering in financial fraud detection for unsupervised learning exist yet. We believe that performance of unsupervised learning models can be improved when using the selected important features based on our framework;

The rest of this paper is organised as follows. In Section 2, we review the techniques and recent development of feature engineering in general study and for financial fraud detection. In section 3, we describe about a real-life dataset from a European bank. Then, in Section 4, we present our development of new framework to create and evaluate effective features for fraud detection model. Afterwards, the experimental composition and the results is shown in Section 5. Finally, conclusion and discussion of the paper are given in Section 6.

2. RELATED WORKS

This paper is closely related to the recent literature on a fraud detection framework that incorporates feature engineering methods. One frequently used feature engineering approach combines two or more features from original data into new ones to represent customer's behaviour on transaction. J.M.Kanter et al [26] developed a cross domain framework that generalises three parts of features, which are Label, Segment, Featurise (L-S-F), to customise the process of feature creations. This feature engineering framework is a general concept to improve an accuracy of machine learning models. Y.Lucas et al. [19] built a conceptual framework of generating history base features using Hidden Markov Models (HMM). The framework calibrates the similarity between an observed sequence and the sequences of past fraud transactions inspected for the cardholders. These examples of feature engineering framework in the financial field are for supervised learning algorithms such as Decision Tree (DT), Random Forests (RF) and Logistic Regression (LR), while Nargesian et al. [8] and Heaton [9] introduce the frameworks for improving an accuracy of unsupervised learning algorithms: Deep Learning (DL), Recursive Neural Network (RNN) and Convolutional Neural Network (CNN) as credit card fraud detection models. The framework for unsupervised learning algorithms applies mathematical functions on a single feature in original data to create new features for improving an accuracy of fraud detection models. Xinwei et al. [6] developed a fraud detection system that uses a progressive feature engineering process based on "Homogeneity-oriented behaviour analysis (HOBA) using a deep learning model. HOBA uses four categories: Recency, Frequency, Monetary value, and Location, to categorise into some small groups based on the similar characteristic on transactions. These papers demonstrate the effectiveness of using feature creations for prediction models.

Feature creation methods in financial fraud detection are roughly divided into two categories: feature aggregation and feature transformation. The aggregated features are used for observing user's behaviour in transactions. Y.Xie et al. [20] developed a rule-based feature engineering method for credit card fraud detection that considers both individual behaviour and group behaviour, and creates group features that classify regular and fraudulent transactions. C.Whitrow et al. [21] introduced the new feature aggregation technique for credit card fraud detection that calculates over transactions observed by a fixed time window and between maximum and minimum amounts. Bahnsen et al. [2] created aggregated features by applying the statistical function of the von Mises distribution on interval time between the last transaction and the latest transaction by each individual customer.

Feature transformation transforms the original features into new ones to describe the original data. The methods of feature transformation applying mathematical functions such as log, square, normalization, addition, subtraction, multiplication, division, mean and standard deviation on each attribute in a dataset are utilised in our framework and these methods are shown the effectiveness of improving an accuracy of machine learning models in general feature engineering studies [8, 9, 25, 27]. For example, J.M.Kanter et al [27] developed the Deep Feature Synthesis algorithm to create features for relational datasets. The algorithm observes relationships in the data and then sequentially applies mathematical functions among the data. Other feature transformation methods in the field of financial fraud detection are for unsupervised learning algorithms including deep learning [6, 22, 23, 24], and they show a high level of effects for unsupervised learning models.

Another feature engineering approach is to select significant features for specific ML algorithms. Lee et al. [4] use a feature selection method for unsupervised learning in credit card fraud detection to select relevant features to a target and they use feature selection methods such as filter, wrapper and embedded. Brodley et al. [10] employ the Expectation-Maximization clustering method that disperse separability and maximum likelihood. Xinwei et al. [6] select relevant features using Chi2 technique in feature selection for classification of e-commerce websites. D. Varmedja et al. [5] concluded that feature selection and balancing unbalanced label dataset should be carried out to enhance a credit card fraud detection for machine learning algorithms. Through the whole results of experiments using the selected features presented that feature selection is remarkably significant in achieving meaningful results.

These studies show the importance of feature selection by a comparison of the performances between ML models built with selected features and other ones built with only original features. Though many studies of feature engineering have proven the effectiveness of feature creation and feature selection individually, they seldom implement both methods together in one framework. In this paper, we use feature engineering methods of feature creation process and feature selection process jointly for ML in financial fraud detection.

3. ONLINE BANKING DATA ON TRANSACTIONS

An online payment dataset is provided by a European bank to verify the effect of the framework and it contains approximately 29,000 transactions across about 2,692 account holders in 3 days. The ratio of fraud labels is about 7% of all transactions. This dataset is partially extracted from over 100,000 transactions for a tentative experiment. In future work, we will examine with the full of transactions after verifying the effect of the framework in this paper.

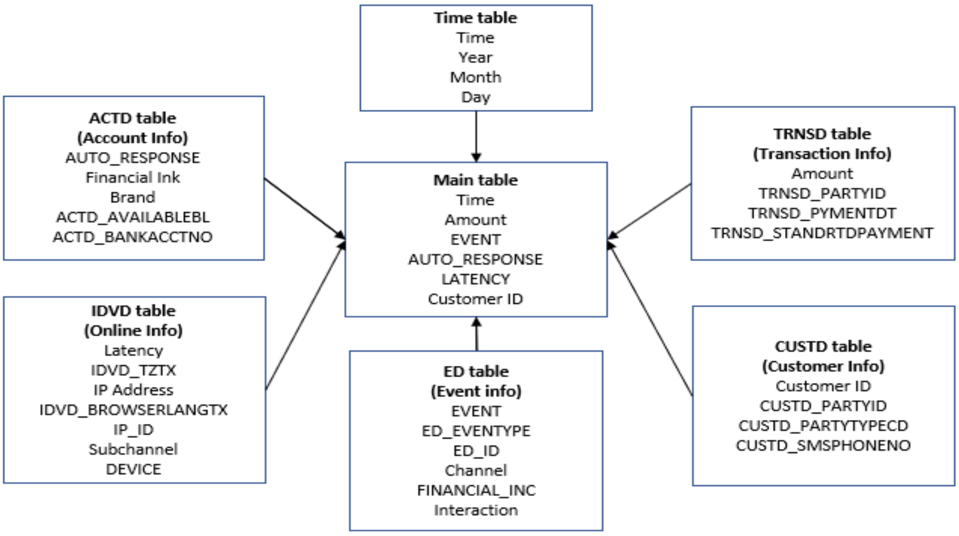

Figure 1. data modelling

The dataset, which is integrated from different tables such as time, account, online, customer's info, transaction, events, is as shown. Descriptions of each feature in the dataset are described in Table 1.

Table 1: Description of Original Features

Attributes	Description	Attributes	Description
ACTD_BANKACCTNO	Account's bank account number	CUSTD_ SMSPHONENO	SMS phone number
ACTD_AVAILABLEBL	Available balance	LATENCY	Latency
TRNSD_FASTER STANDARDPAYMENTIND	Faster or Standard payment indicator	IP Address	Access IP Address
ED_EVENTTYPETX	Type of payments	Interaction	Internet banking, branch, mobile, Tel
Customer ID	Customer Party ID	Time	Access date time / Timestamps
EVENT	Event of transaction	Financial INC	Transfer bank name
IDVD_INTESSIONID	Internet Section ID	Brand	Financial Institute name
IDVD_TZTX	Time zone of transaction	Sub channel	Sub-channel name
IDVD_USERAGE0TTX	Online user agent	DEVICE	Access devices
AUTO_RESPONSE	Auto response	IP_ID	Online banking ID

4. Feature Engineering Framework for Financial Fraud Detection

The main contribution of our framework is to join two processes of feature creation and feature selection illustrated in Figure 2.

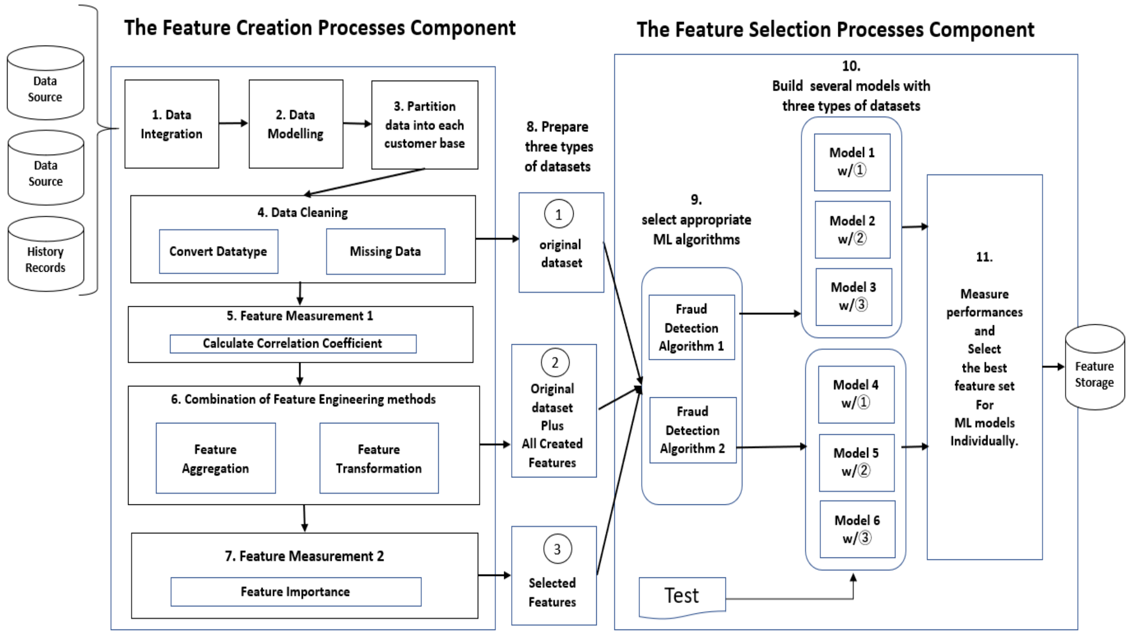
The Feature Creation Processes Component
The Feature Selection Processes Component
Data Source
Data Source
History Records
1. Data Integration
2. Data Modelling
3. Partition data into each customer base
4. Data Cleaning
Convert Datatype
Missing Data
5. Feature Measurement 1
Calculate Correlation Coefficient
6. Combination of Feature Engineering methods
Feature Aggregation
Feature Transformation
7. Feature Measurement 2
Feature Importance
8. Prepare three types of datasets
① original dataset
② Original dataset Plus All Created Features
③ Selected Features
9. select appropriate ML algorithms
Fraud Detection Algorithm 1
Fraud Detection Algorithm 2
10. Build several models with three types of datasets
Model 1 w/①
Model 2 w/②
Model 3 w/③
Model 4 w/①
Model 5 w/②
Model 6 w/③
Test
11. Measure performances and Select the best feature set For ML models Individually.
Feature Storage

Figure 2. Feature Engineering Framework for Fraud Detection Models.

4.1. Feature Creation Processes

In the feature creation component, there are seven steps to create feature candidates and measure important features. The raw data collected from various sources is a mess and needs to be cleaned by dealing with data formats and missing values before implementation of feature engineering. The processes from step 1 to step 5 are relevant to pre-processing feature engineering, specifically in step 5, similar attributes are removed from original data to avoid over fitting by using correction coefficient as an evaluation method.

(a) Feature Aggregation based on Customer Behaviour

Feature aggregation represents customer's behaviour on online transaction. The original data is grouped by each customer ID to build an individual customer's profile. Aggregation makes more detailed features that express the individual customer's regular patterns by combining two or more attributes from various tables as shown in Figure 3 below.

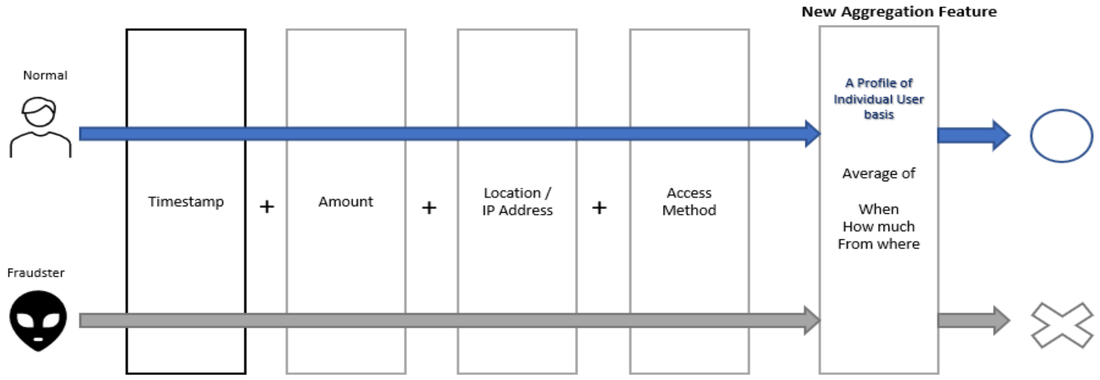

Figure 3. Image of Combining Multiple Features

In Table 2 describes some examples of feature aggregation that enable ML algorithms to learn various patterns of customer' behaviour and to classify a fraud pattern more easily.

Table 2. Feature Aggregation

Attributes	Combinations
Time	Days since the last transactions Hours since the last transactions Minutes since last transactions Days since the last access by same device Hours since the last access by same device Minutes since the last access by same IP address Hours since the last access by same IP address Days since the last event type occurred Hours since the last event type occurred Days since the last transaction occurred from specific location/ATM Hours since the last transaction occurred from specific location/ATM

IP Address	IP address of access device since last transaction IP address of access device since last transaction
Amount	Amount of the last transaction Amount of the last transaction from specific location/ATM Amount of the last transaction via IP address
Channel	Channel type when each event is occurred
Event Type	Event type accessed via IP address Event type accessed by a specific device

(b) Feature Transformation based on mathematical functions

We selected several mathematical functions to transform a single feature to different aspects. Some examples of mathematical functions used for transformation features are shown in Table 3.

Table 3. Feature Transformation

Examples of Mathematical Functions	Formula/Equations
Confidence Interval	a statistic estimation formula that uses the normal distribution for observing a point estimate by calculating ***maximum***, ***minimum***, ***median***, and ***mean***
Standard Deviation	a method of scaling the values based on z-score which calculates the following equation. $Z=(x-\mu)/\sigma$ where x:value to be transformed, μ: mean value of the data, σ: standard deveation
Binning	a way to group figures of continuous numbers into bins
Clustering (K-Means)	a way to group a set of spots into clusters based on a distance measure. Customer's info can be classified with the distances from an actual and some groups based on similar data patterns by using k-means
Linear	The equation: Let A1, ……, An be n matrices having dimension K x L. $B = \alpha_1 A_1 + \dots + \alpha_n A_n$
Logarithm	Log transformation is one of the popular transformation. $X'_I = \log(xi)$

Now, we created approximately 42 feature candidates in the real-life dataset using feature aggregation and feature transformation methods as described in Table 4.

Table 4. New created features based on aggregation and transformation

Feature Engineering Time Series	Description
Year	Transaction year
Month	Transaction month
Day	Transaction day
Hour	Transaction hour

Minute	Transaction minute
Second	Transaction second
Weekday	Transaction weekday

Day of year	Days of year from transaction
Feature Engineering **Clustering**	**Description**
Class	Clustering group by k-means based on customer characters
Aggregations based on customer behaviour	**Description**
Customer ID conf Rate	Attributed rate scale by confidence on customer ID
ED_EVENT conf Rate	Attributed rate scale by confidence on Event Type
Action Type conf Rate	Attributed rate scale by confidence on Action Type
DEVICE conf Rate	Attributed rate scale by confidence on Device frequency
Amount conf Rate	Attributed rate scale by confidence on Amount
Customer ID EVENT par Day	Group by customer ID and Event frequency per day
Customer ID IP Address par Day	Group by customer ID and IP address frequency per day
Customer ID DEVICE par Hour	Group by customer ID and device frequency per hour
Customer ID USER count Minute	Group by customer ID and user agent counts per minute
Customer ID Channel count Minute	Group by customer ID and channel counts per minute
Customer ID counts	Count each customer ID
New feature	Time to next transaction for each customer
Transformations based on mathematical method	**Description**
Latency diff	Difference Latency
Amount diff	Difference Amount
Day diff	Difference Day
Hour diff	Difference Hour
Minute diff	Difference Minute
Access min	Minimum access time
Access max	Maximum access time
Access std	Standardization of Access time
LATENCY std	Standardization of Latency
Amount std	Standardization of Amount
Amount log	Log Transform of Amount
Min log	Log Transform of Minute
Sec log	Log Transform of Second
Day bin	Binning of Day
Min bin	Binning of Minute
Channel Event	Linear combinations (Channel and Event Type)
Action IP	Linear combinations (Action type and IP address)
Event Latency	Linear combinations (Event and Latency)
Event Sub Device	Linear combinations (Event Type and subchannel and device)
Event INC Code	Linear combinations (Event Type and Auth code and FC type)

Day of year	Days of year from transaction
Feature Engineering Clustering	**Description**

Class	Clustering group by k-means based on customer characters
Aggregations based on customer behaviour	**Description**
Customer ID conf Rate	Attributed rate scale by confidence on customer ID
ED_EVENT conf Rate	Attributed rate scale by confidence on Event Type
Action Type conf Rate	Attributed rate scale by confidence on Action Type
DEVICE conf Rate	Attributed rate scale by confidence on Device frequency
Amount conf Rate	Attributed rate scale by confidence on Amount
Customer ID EVENT par Day	Group by customer ID and Event frequency per day
Customer ID IP Address par Day	Group by customer ID and IP address frequency per day
Customer ID DEVICE par Hour	Group by customer ID and device frequency per hour
Customer ID USER count Minute	Group by customer ID and user agent counts per minute
Customer ID Channel count Minute	Group by customer ID and channel counts per minute
Customer ID counts	Count each customer ID
New feature	Time to next transaction for each customer
Transformations based on mathematical method	**Description**
Latency diff	Difference Latency
Amount diff	Difference Amount
Day diff	Difference Day
Hour diff	Difference Hour
Minute diff	Difference Minute
Access min	Minimum access time
Access max	Maximum access time
Access std	Standardization of Access time
LATENCY std	Standardization of Latency
Amount std	Standardization of Amount
Amount log	Log Transform of Amount
Min log	Log Transform of Minute
Sec log	Log Transform of Second
Day bin	Binning of Day
Min bin	Binning of Minute
Channel Event	Linear combinations (Channel and Event Type)
Action IP	Linear combinations (Action type and IP address)
Event Latency	Linear combinations (Event and Latency)
Event Sub Device	Linear combinations (Event Type and subchannel and device)
Event INC Code	Linear combinations (Event Type and Auth code and FC type)

4.2. Feature Selection Processes

Three types of datasets are set up after the processes in the feature creation component. The first dataset is original features, the second one is a set of original features and created features in the feature aggregation and transformation processes. The last dataset is only selected features from the second one based on feature importance. In the feature selection component, any ML algorithms for fraud detection can be chosen according to user's needs. In the framework, weselected two ML algorithms of support vector machine (SVM) and isolation forest (IF). SVM is a supervised learning algorithm and popularly used for fraud detection in many studies

[3,11,12,13]. In their studies, performance of SVM is steady and fine. IF is an unsupervised learning algorithm and works well for anomaly detection [14,15,16]. These ML algorithms use the three datasets individually to build each model and evaluate their results based on classification report and AUC. Eventually, the best feature sets can be selected for each ML model.

(a) Feature Importance

As an evaluation method of relevant features, we select feature importance from RF model to measure the relative importance of each input feature. Scores of feature importance are calculated by the training data used to the model. In the RF model, every node indicates a status of how to split values in an individual feature. The status is based on impurity, which is Gini impurity or information gain (entropy) in case of classification. While training the RF model, feature importance of each feature is computed how much a single feature contributes to reducing the weighted impurity. The figure 4 describes feature importance of each feature in the second dataset. It indicates that many importance features with high scores are the created features by feature engineering methods. Following this evaluation result, we selected 46 features out of 66 features in the second dataset.

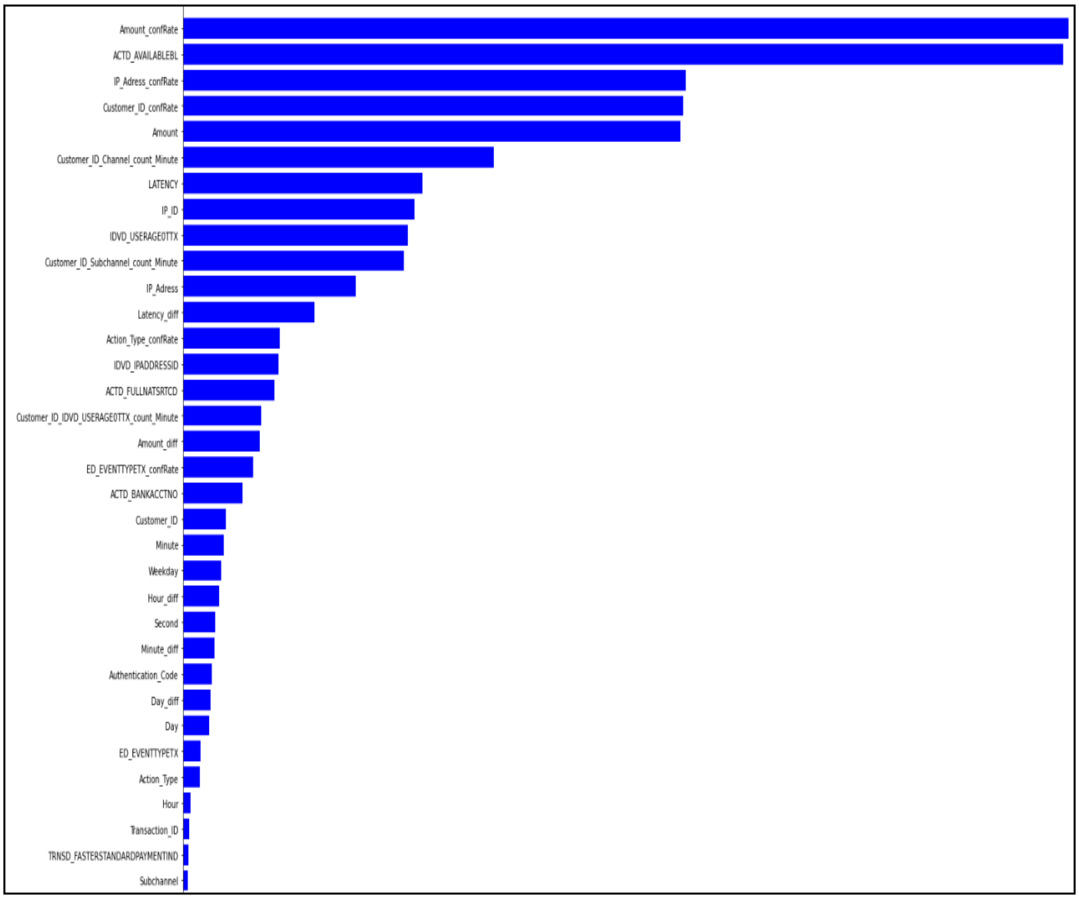

Figure 4. Feature Importance

(B) Fraud Detection Algorithms

- **Support Vector Machine**

Support vector machine (SVM) is a supervised learning algorithm and a popular classification method in financial fraud detection [3,11,12,13] to group values in dataset by applying a boundary line, called a hyper plane, which segregates a fraud pattern from normal patterns[18]. The best boundary will be determined by finding a hyper plane where splits the two classes of data locations by calculating maximum distance between the two classes shown in figure 5. A hyper plane is defined by the following function [18],

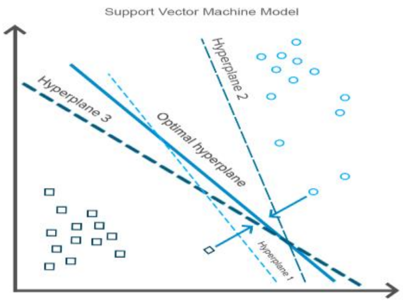

Figure 5. support vector machine approach

Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right) \qquad (1)$$

Constraints:

$$\begin{aligned} & y_i - wx_i - b \le \varepsilon + \xi_i \qquad (2) \\ & wx_i + b - y_i \le \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \ge 0 \end{aligned}$$

Linear SVM: (3)

$$y = \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right)\cdot\left\langle x_i, x\right\rangle + b$$

(4)

- **Isolation Forest Algorithm**

Isolation forest (IF) is an unsupervised learning algorithm for anomaly detection [14,15,16] and consists of multiple isolation trees which are created by repeating swiftly and randomly selecting attributes between the maximum and minimum values. Attributes values of anomalous instances are commonly different from the regular instances. The median depth of the instance in the forest which is consisted of multiple isolation trees is calculated to give a measure of the normality and anomalous scores of the instance. Equation of the algorithm is described as following:

$$Anomaly\ Score\ (S) = 2^{\frac{-E(h(k,m,N))}{c(n)}}$$

$$,where\ c(n) = 2(\ln(n-1) + 0.5772156649) - 2\left(\frac{n-1}{n}\right)$$

$$,where\ n\ is\ a\ number\ of\ data\ points\ in\ a\ chosen\ sample$$

$$,where\ E(h(k,m,N)) = \frac{\sum_{i=1}^{N} \begin{cases} if\ k == 1, \sum_{j=1}^{M} 1 \\ else,\ \sum_{j=1}^{M} 1 + c(k) \end{cases}}{N}$$

$$,where\ N\ is\ a\ total\ number\ of\ trees$$

$$,where\ M\ is\ a\ total\ number\ of\ binary\ splits$$

$$,where\ k\ is\ a\ total\ number\ of\ data\ points\ in\ the\ final\ node\ (exit\ node)$$

Equation 1. Calculation in isolation forest

Anomaly scores are calculated by the average cross multiple trees in the forest. In figure 6 and figure 7 show each sub dataset that was split randomly and the isolated data point of a non-anomalous point and an anomalous point [17].

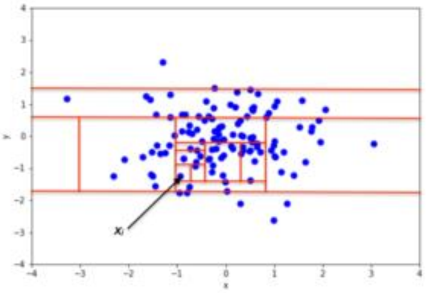

Figure 6. Isolated data point of a non-anomalous point [17]

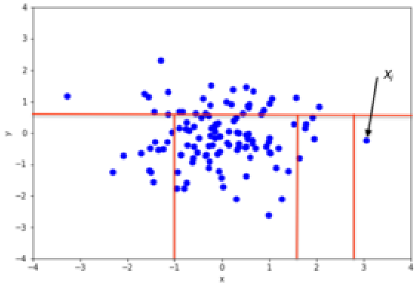

Figure 7. Isolated data point of an anomalous point [17]

5. MODELLING AND RESULTS

In the experiment of the feature engineering framework, six different models based on SVM and IF techniques are developed with three different types of feature sets, which are only original features, all created features and original features, selected features based on feature importance shown in Table 5. And subsequently, their performance is analysed and compared. Under Jupiter Notebook, python with sklearn library is used to create and evaluate features, and build SVM and IF models. As the performance evaluation methods, we use AUC and a classification report including precision, recall and F1-score. Each measurement is proceeded depends on how many target variable of fraud flag ("1") is correctly detected by each model.

Table 5. Selected Features from All features in the dataset

Selected Features	Description
ACTD_AVAILABLE	Available balance
ACTD FULLNATSRTCD	Available transfer code
ACTD BANKACCTNO	Available bank account
Amount conf Rate	Attributed rate scale by confidence on amount
Latency diff	Difference Latency
Latency	Latency
Event Latency	Event latency
Event Act	Event action
Event INC Code	Event Inc code
IDVD USERAGETTX	Online user agent
Sub Channel PERSONAL	Sub channel type
Action IP	Action IP
Action Type conf Rate	Attributed rate scale by confidence on Action type
Amount	Transaction amount
Minute	Transaction minute
Hour	Transaction hour
Day	Transaction day
weekday	Transaction weekday
Customer ID IDVD USERAGE count Minute	Group by customer ID and online user agent frequency per minute
Customer ID Channel count Minute	Group by customer ID and channel frequency per minute
Customer ID counts	Group by customer ID counts per day
Amount diff	Difference Amount
Device DIGITAL	Access device and access type
ED EVENT TYPETX conf Rate	Attributed rate scale by confidence on event type
Minute diff	Difference Minute
Hour diff	Difference Hour
Day diff	Difference Day
Transaction ID	Transaction ID

Table 6: Performance of each model using three types of feature sets

Classifiers	F1-Measure	Precision	Recall	AUC
SVM with original data (1)	0.73	1.0	0.57	0.79
IF with original data (1)	0.25	0.24	0.26	0.59
SVM with all features (2)	0.97	1.0	0.94	0.97
IF with all features (2)	0.40	0.39	0.42	0.68
SVM with selected features (3)	0.95	1.0	0.91	0.95
IF with selected features (3)	0.60	0.57	0.62	0.79

* () ...dataset type

The measurement results of ML models using different feature sets are shown in Table 6. Recall shows the proportion of the actual fraud actions that were accurately detected, while precision donates the proportion of the accurately detected fraud actions to the detected fraud actions. Specifically, the aspect of F1-measure and AUC estimate the overall performance of ML models.

By comparing performances of the ML models using engineered features created by our framework with the ML models using only original features, all ML models using engineered features improve the accuracy in every measurements by nearly 40% on the F1-measure and 20%

on the AUC value. The SVM model using all features achieves the highest F1-measure of 0.97 and the highest AUC of 0.97, while the SVM model using only original data records the F1-measure of 0.73 and the AUC of 0.79. The IF models using created features through our framework have much better F1-measure scores of 0.60 and AUC of 0.79 than the IF model using original data that has the scores of 0.25 on F1-measure and 0.59 on AUC.

We compare the effectiveness of the feature set using all created features with using selected features based on feature importance to evaluate the compatibility between the effective feature set and a specific ML algorithm. The performance of SVM model using all features is better than SVM model using the selected features, whereas the performance of IF model using selected features is better than IF model using all features. The AUC value of SVM model using all features becomes 0.97, whereas the AUC value of SVM model using the selected features is 0.95. The AUC value of IF model using all features becomes 0.68, whereas the AUC value of IF model using the selected features is 0.79. We conclude that the important feature set is not effective for any ML algorithms in common. Finally, by comparing the performance of unsupervised learning models with supervised learning models, the AUC values and F1-measure scores of supervised learning models are higher than unsupervised learning models in every measurements. Overall, the results above demonstrate the effectiveness of the proposed feature engineering framework.

6. Conclusions and Future Work

In this paper we have proposed a new framework of feature engineering for ML models in financial fraud detection. What distinguishes our framework from others is that it involves both feature creation and feature selection. In addition, our feature creation process puts together two types of feature creation: feature aggregation and feature transformation. Moreover, our feature selection process is compatible with a variety of ML algorithms. Hence, our framework is general and applicable to many types of ML algorithms used in financial fraud detection and could enhance the existing financial fraud detection models. Using an actual financial transaction dataset from a private bank in Europe, we have shown that our framework improves the accuracy of ML model prediction significantly 40% on the F1-measure and 20% on the AUC value comparing with baseline models. We would like to conclude the paper with two caveats. First, although our experiment using an actual dataset shows an improvement in ML model prediction, the experiment uses standard ML algorithms such as SVM and IF, our framework will be applicable to richer algorithms such as a deep learning algorithm, which has recently attracted attention in financial fraud detection. Using such an algorithm in our framework is listed on our future work. Second, in our experiment, the data are limited to a small subset of large amounts of transactions. It would enhance fraud detection further if more contextual data about customer behaviour and transactions via various devices or online websites are used in our framework. Despite these caveats, we hope that our proposed framework will be useful for financial institutions to fight against financial fraudulent activities

References

[1] Financial Fraud Action UK. January to June 2020 fraud update: Payment cards. Remote banking and cheque s.1.: Financial Fraud Action UK, 2020.

[2] A.C.Bahnsen, D.Aouada, A.Stojanovic & B.Ottersten (2016) “Feature engineering strategies for credit card fraud detection”, Expert Systems With Applications, Vol. 51,P134-142.

[3] A.Yesilkanat, B. Bayram, B.A.Koroglu & S.Arslan (2020) “An Adaptive Approach on Credit Card Fraud Detection Using Transaction Aggregation and Word Embeddings”, Semantic Scholar, corpus ID:218980594

[4] H.Lee, D.Choi, H.YIM, E.Choi, W.Go, T.Lee, I.Kim & K.Lee (2018) "Feature Selection Practice For Unsupervised Learning of Credit Card Fraud Detection", Journal of Theoretical and Applied Information Technology, Vol.96, No2, P408-417

[5] D.Varmedja, M.Karanovic, S.Sladojevic, M.Arsenovic & A.Andrela (2019) "Credit Card Fraud Detection - Machine Learning methods", 18th International Symposium INFOTEH-JAHORINA, 20-22 March 2019

[6] Z.Xinwei, H.Yaoci & W.Qili (2019) "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture", Elsevier Information Sciences,online press

[7] S.Wang, J.Tang & H.Liu (2016) *Feature Selection*, Encyclopaedia of Machine Learning and Data Mining, P1-9, Springer Link

[8] F.Nargesian, H.Samulowits, U.Khurana, E.B.Khalil & D.Turaga(2017) "Learning Feature Engineering for Classification", IJCAI, P2529-2535

[9] J.Heaton (2017) "An Empirical Analysis of Feature Engineering for Predictive Modeling", Cornell University arXiv org, 1701.07852v1

[10] J.G.Dy & C.E.Brodley (2004) "Feature Selection for Unsupervised Learning", Journal of Machine Learning Research Vol. 5, P845-889

[11] C.Wang & D.Han (2018) "Credit card fraud forecasting model based on clustering analysis and integrated support vector machine", 406(1), P13861-13866, Springer Link

[12] Y.Jain, N.Tiwari, S.Dubey & S.Jain (2019) "A Comparative Analysis of Various Credit Card Fraud Detection Techniques", International Journal of Recent Technology and Engineering, Vol7, P2277-3878

[13] M.Khedmati, M.Drfani & M.GhasemiGol (2020) "Applying support vector data description for fraud detection", Cornell University arXiv org, 2006.00618v1

[14] S.P.Maniraj, A.Saini, S.D.Sarkar & S.Ahmed (2019) "Credit Card Fraud Detection using Machine Learning and Data Science", International Journal of Engineering Research & Technology, Vol. 8, P2278-0181

[15] F. Carcillo, Y.L.Borgne, O.Caelen, Y.Kessaci, F.Oble & G.Bontempi (2020) "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection", ResearchGate, DOI:10.1016

[16] F.Liu, K.M.Ting & Z.H. Zhou (2008) "Isolation Forest", 8th IEEE International Conference on Data Mining: P413-422

[17] Z.Ding & M.Fei (2013) "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window", Vol. 46, P12-17

[18] S.Patel & S.Gond (2014) "Supervised Machine Learning for Credit Card Fraud Detection", International Journal of Engineering Trends and Technology (IJETT), Vol. 8

[19] Y.Lucas, P.E.Portier, L.Laporte, L.H.Guelton, O.Caelen, M.Granitzer & S.Calabretto (2020) "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs", ScienceDirect, ELSEVIER,1909.01185v1

[20] Y.Xie, G.Liu, R.Cao, Z.Li, C.Yan & C.Jiang (2019), "A Feature Extraction Method for Credit Card Fraud Detection", 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), 2019.00019, DOI 10.1109

[21] C.Whitrow, D.J.Hand, P.Juszczak, D.Weston & N.M.Adams (2008), "Transaction Aggregation as a Strategy for Credit Card Fraud Detection", Data Mining and Knowledge Discovery, Vol. 18, P30-55

[22] K.Fu, D.Cheng, Y.Tu & L.Zhang (2016), "Credit Card Fraud Detection Using Convolutional Neural Networks", International Conference on Neural Information Processing, P483-490

[23] S.Misra, S.Thakur, M.Ghosh & S.K.Saha (2019), "An Autoencoder Based Model for Detecting Fraudulent Credit Card Transaction", ScienceDirect, Vol. 167, P254-262

[24] J.Jurgovsky, M.Granitzer, K.Ziegler, S.Calabretto, P.Portier, L.Heguelton & O.Caelen (2018), " Sequence Classification for Credit-Card fraud detection", Expert Systems with Applications

[25] F.Nargesian, H.Samulowitz, U.Khurana, E.B.Khalil & D.Turaga (2017), "Learning Feature Engineering for Classification", 26th International Joint Conference on Artificial Intelligene, (IJCAI-17)

[26] J.M.Kanter & K.Veeramachaneni (2016), "Label, Segment, Featurize: a cross domain framework for prediction engineering", IEEE International Conference on Data Science and Advanced Analytics (DSAA), P430-439

[27] J.M.Kanter & K.Veeramachaneni (2015), “Deep Feature Synthesis: Towards Automating Data Science Endeavors”, IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2015.734485

AUTHORS

Chie Ikeda holds a master’s degree in data Analytics and a PhD student at London Metropolitan University, UK. She is also an assistant manager and data scientist at Aflac Life Insurance Company in Japan. One of her responsibilities at the company is to lead her project team to successfully build an AI model for detecting a fraud.

Karim Ouazzane is a currently a full professor of computing and knowledge exchange at London Metropolitan University. He is Director of research and enterprise in the school of Computing and Digital Media and the University Knowledge Transfer Partnership Director. He is currently the Chair of the European Cyber Security Council at Brussels. He worked and acted as a consultant for a number of companies such as Endress-Hauser, ICI, Power Gen, Schlumberger, Barclays, Lifeline IT and Lloyds Banking Group in the UK in the area of machine learning and cyber security.

Qicheng Yu is a senior lecturer at London Metropolitan University. He leads modules in data mining, programming for data analytics, e-business and e-commerce, database and web applications development and supervises PhD research students.

© 2020 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.