New Feature Engineering Framework for Deep Learning in Financial Fraud Detection

Chie Ikeda, Karim Ouazzane, Qicheng Yu, Svetla Hubenova
School of Computing and Digital Media
London Metropolitan University
London, UK

***Abstract*—The total losses through online banking in the United Kingdom have increased because fraudulent techniques have progressed and used advanced technology. Using the history transaction data is the limit for discovering various patterns of fraudsters. Autoencoder has a high possibility to discover fraudulent action without considering the unbalanced fraud class data. Although the autoencoder model uses only the majority class data, in our hypothesis, if the original data itself has various feature vectors related to transactions before inputting the data in autoencoder then the performance of the detection model is improved. A new feature engineering framework is built that can create and select effective features for deep learning in remote banking fraud detection. Based on our proposed framework [19], new features have been created using feature engineering methods that select effective features based on their importance. In the experiment, a real-life transaction dataset has been used which was provided by a private bank in Europe and built autoencoder models with three different types of datasets: With original data, with created features and with selected effective features. We also adjusted the threshold values (1 and 4) in the autoencoder and evaluated them with the different types of datasets. The result demonstrates that using the new framework the deep learning models with the selected features are significantly improved than the ones with original data.**

***Keywords*—Financial fraud; online banking; feature engineering; unbalanced class data; deep learning; autoencoder**

I. Introduction

As the online payment system advances, fraud schemes have shifted from physical fraud actions using ATMs into an advanced technique that uses digital banking accounts. Unauthorized remote banking fraud is formed by three categories: Internet banking, telephone banking and mobile banking. A fraudster accesses a customer's bank account through these remote banking channels and steals money by making an unauthorized money transfer from the account. UK finance announced that total losses through remote banking in the United Kingdom have increased and reached £197.3 million in 2020, 31percent higher than in 2019. The annual number of cases of internet banking fraud and mobile banking fraud has been growing rapidly from 32,721 cases in 2019 to 66,150 cases in 2020. Other financial fraud losses such as payment cards and cheques decreased from £470.2 million to £452.6 million [1].

The fraud Detection System (FDS) used by many financial institutions, has not caught up with the advancement in fraudulent schemes on remote banking. To address constant changes in fraud behavior, some financial industries employ machine learning (ML) methods in FDS [2, 3], but it is still challenging to reveal new fraudulent behaviors by applying ML to raw data only.

Financial transaction data is also very unbalanced because legitimate transactions account for 90% and above of all transaction data and only less than 10% of the rest of the data is fraud. It is difficult to find fraudulent patterns out for ML algorithms specifically for supervised learning.

In the new feature engineering framework published in [19], we created and selected the effect features for fraud detection models built with ML algorithms: We selected Support Vector Machine (SVM) and Isolation Forest (IF) as fraud detection models.

Throughout this research, we apply the feature engineering framework on remote banking data for deep learning with the experimental dataset being provided by a European private bank.

Deep learning has been popularly used for image, audio and video recognition in terms of coping with big data in depth. It learns by dividing input data into a plurality of segmented data patterns through many hidden layers. Recently, it came to be used for classification issues such as fraud detection in the financial area. The original concept of deep learning will be traced to studies of artificial neural networks (ANN). Autoencoder is a type of ANN, an unsupervised deep learning algorithm [4]. It learns how to compress and decompress input data for representation of the original input data and consists of three layers: Encoder, latent (hidden) layer, decoder (Fig. 1). It discovers specific features from the given data during the process of data compression, also known as dimensionality reduction, and how to map the compressed features to the latent layer. The autoencoder finds out how to reconstruct the input data from mapping the features. The most advantage of using autoencoder for financial fraud detection is that autoencoder does not need fraudulent transaction data to learn fraud patterns. As mentioned above, the proportion of fraud transaction data is very little whereas the number of legitimate transaction data is very large. It is difficult to keep track of new fraudulent behavior and state-of-the-art fraud schemes from a few fraud samples because fraudulent actions are not carried out by one person. On the other hand, legitimate transactions are carried out by the same customer who holds his or her own bank account or credit card. Autoencoder can reconstruct customers' behavior patterns by learning from

specific features among large history transaction data. Autoencoder models judge fraudulent data by using loss function with mean squared error (MSE) that measures the error distance of variables in specific features between the learnt data and new input data. There are some related studies of fraud detection using the autoencoder model [5, 6, 7, 8] and they chose autoencoder techniques from the perspective of coping with unbalanced transaction datasets. They commonly use two popular techniques of feature engineering, which are principal component analysis (PCA) and standardization. PCA is a technique of dimensionality reduction and uses orthogonal transformation that computes covariance matrix which represents the correlation between two variables. Unlike machine learning models, deep learning is essential for data processing standardization as it standardizes and weight each attribute to measure how much specific features influence.

Standardization is an essential data processing for using deep learning because deep learning multiplies each attribute and sets the weighting coefficients. Deep learning has not implemented feature engineering on input data from the point of view of adding latent data patterns.

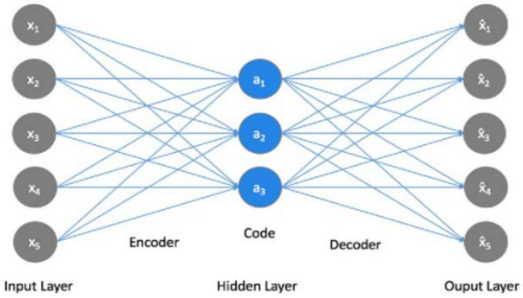

Fig. 1. Autoencoder with Hidden Layer.

In this paper, we propose a new feature engineering framework that newly creates features using feature engineering methods of feature aggregation and feature transformation and selects effective feature candidates for deep learning. In the experiment, the autoencoder is used for building a fraud detection model and verifying the effect of the paper. The rest of the paper is structured as follows: Section 2 reviews the recent development of feature engineering in financial fraud detection for deep learning. Section3 develops a new feature engineering framework that combines feature creation and feature selection processes, and section 4 introduces an autoencoder for fraud detection. Section 5 presents the experimental remote banking dataset, and the final section demonstrates a simulation of the framework for the deep learning model. Finally, section 6 presents results, discussions, and future work.

The main contribution of this study is around improving the accuracy of fraud detection models through using the engineered features produced with the framework, which are the combination of creation and selection processes.

II. Related Work

Until several years ago, the credit card was the great majority of transaction methods in financial services at ATMs, shops, and online shopping. In recent years, remote banking has also become a popular method for transferring money and at the same time, financial fraud losses through remote banking exceed fraud losses of using credit cards according to a report by UK finance in 2021 [1]. Despite the increase of the fraudulent cases of remote banking, there are still very few studies on using feature engineering for deep learning in remote banking.

A. Feature Engineering Framework for Financial Fraud Detection

There exist some similar works of feature engineering framework for financial fraud detection in [9,10,11,12] and they all use feature aggregation methods to create behavior attributes that reveal latent fraudulent patterns. J.S. Kalwihura et al. [11] and Zhang et al. [10] use the HOBA feature engineering methodology which groups into homogenous fraudulent patterns by using feature aggregations based on recency, frequency and monetary (RFM) for insurance fraud detection. The RFM is for behavior analysis which is popularly used in the marketing area. Feature aggregation methods in HOBA feature engineering consist of four aggregation categories related to behavior analysis based on a defined period during a transaction. HOBA also comprises a feature selection method which is a bootstrapped ensemble of bagged trees to select a subset of features from original data. They select random forest as an experimental model which demonstrates a 56.2% increase in the F1-score compared against the original data. Y. Lucas et al. [9] suggest using a feature engineering framework based on multi-perspective Hidden Markov Models (HMMs) for credit card fraud detection. The history of credit card transactions has the card holder's habits of the timing or the place of using a credit card in the last 24h. HMM is a sequence classification model which considers the sequential properties of transaction data. The multi-perspective HMMs categorize a symbol on transactions such as "merchant and amount", "timing", "fraud or customer"," genuine" and observe each symbol as the sequential event on transactions. The HMMs calculate the likelihood of sequences of observed symbols and create features of each event. To measure the effectiveness of the addition of the HMM features, they use perspective, recall and AUC metrics, and random forest as an experimental model. Consequently, the use of the HMM-based features improved the precision-recall AUC of the random forest model significantly compared with the use of the original features only. All the above studies have demonstrated the impact of using feature engineering methods on data with improved performance of machine learning models. In their works, they focused only on the feature aggregations side to reveal latent fraudulent patterns. A. Nagaraja et al. [14] introduced an approach for any network anomaly detection using feature transformation based on mathematical methods. They use feature clustering based on the Gaussian distribution function and a k-Nearest Neighbours (KNN) classifier as a detection model for finding the similarity between observations. The distribution function provides the equivalent deviation and threshold values to carry similarity calculation, and then the distance function of KNN measures the distance of the transformation features and determines if the input is fraud or a legitimate value. Using transformation features improves the detection accuracy in comparison with using the raw data only.

In the new framework [19], we use feature aggregation and feature transformation jointly to create important feature candidates. R. Wedge et al. [15] suggest Deep Feature Synthesis (DFS) that creates new attributes for machine learning models of credit card fraud detection using the relational structure of the dataset. In the processes of DFS, both feature aggregation and transformation methods are used to create new features using attributes of the related transactions. For instance, they applied the Hour in transaction time to determine when a transaction has occurred during the day and use statistical methods i.e., average, mean, sum and standard deviation to express the user behavior on the transaction time base. Timestamps in transactions are significant processes in DFS to compute features of every month and within 24 hours. Eventually, they generated 237 features (over 100 behavioral pattern features) for each transaction and reduced the false positive rate by 54%. However, in their study, they use all 237 generated features which may cause overfitting if the number of the training data is not enough.

In the work described in [19], we used the engineered features created through using the feature engineering framework which has improved the performance of machine learning models.

In this paper, we make use of the new feature engineering framework for deep learning, specifically for autoencoder neural network models.

B. Fraud Detection using Autoencoder Neural Network

Fraud transaction data is always imbalanced and needs to be carefully handled while using machine learning algorithms. Popular methods of coping with imbalanced datasets are oversampling and undersampling which are techniques to balance the class distribution. Oversampling is utilized to synthesize new samples of fraudulent classes but, it will take in noise. Undersampling removes samples from the majority class in the trained dataset but, it may remove useful information or important data. Autoencoder is good for coping with imbalanced datasets without considering the minority class issue because it only uses majority class samples. Some research for credit card fraud detection uses an autoencoder model [16, 17, 18]. P. Jiang et al. [16] designed a six-layer autoencoder for the dataset and selected SoftMax with cross-entropy as the loss function for final classification to detect credit card fraud. The autoencoder model improved the classification accuracy of the fraud class when the threshold was equal to 0.6. A. Pumsirirat et al. [17] used deep learning based on auto-encoder and restricted Boltzmann machine for credit card fraud detection because fraudsters gain new technology that enables them to steal money from customers. Their autoencoder applied backpropagation by setting the input data equal to the output data. Restricted Boltzmann machine can reconstruct legitimate transactions to discover fraudsters from legitimate patterns and holds two layers, input layer and hidden layer. They used the library of TensorFlow to implement autoencoder and restricted Boltzmann machine. The number of studies of financial fraud detection using autoencoder is not a few, but almost all studies use only raw data as the input data for autoencoder. They do not apply feature engineering methods to the raw data.

III. Feature Engineering Framework for Deep Learning Model

The main contribution of the new framework lies in joining two processes of feature creation and feature selection (Fig. 2). Whether machine learning or deep learning algorithms are selected, the processes of feature creation and feature selection in the framework remain the same. In the case of feature creation for deep learning, it is necessary to standardize variables in all features before building a model.

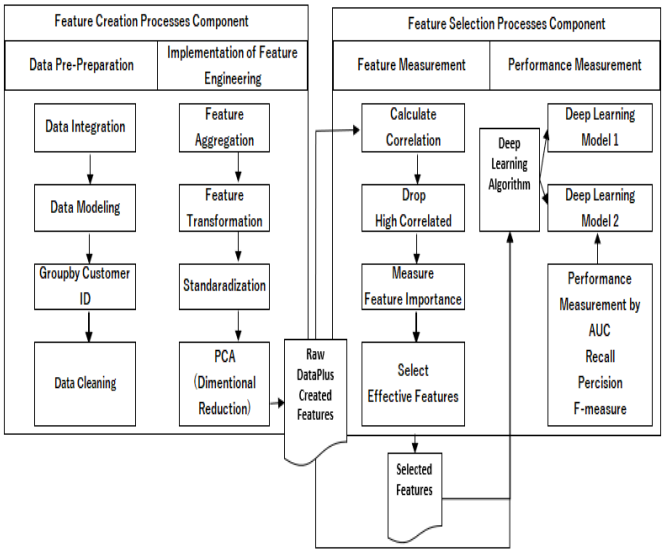

Fig. 2. Feature Engineering Framework.

A. Feature Creation Processes

In the feature creation component, there are two categories: data preparation and implementation of feature engineering. Data preparation contains four processes before the data are ready for implementation using feature engineering methods. The raw data collected from various sources are not clean and need to be maintained by handling missing data and unifying data formats. After the data preparation is made, feature aggregation and feature transformation are sequentially carried out. At this point, a first feature set candidate including all features which include newly created features and original attributes are prepared.

1) Feature aggregation based on customer behavior: Feature aggregation represents a customer's behavior when an online transaction occurs. Based on a unique customer ID, some action attributes e.g., amount, time, access device and network information are aggregated. Aggregation increases the dimensions that can express the data pattern in more detail. The created features by these aggregations represent latent customers' behavior from various angles of data. Table I describes some attribute candidates which can be aggregated with other action attributes for creating individual customer's journeys via online banking.

TABLE I. FEATURE AGGREGATION

Attributes	Combinations
Time	- Days since the last transactions - Hours since the last transactions - Minutes since the last transactions - Days since the last access by same device - Hours since the last access by same device - Minute 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 transactions occurred from specific location/ATM
IP Address	- 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 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

2) Feature transformation based on mathematical functions: There are some available mathematical functions and equations to transform a single attribute into other dimensions by mapping data. The purpose of using transformations is to generate features that discover implications in a given data from mathematical functions i.e., scaling (standardization), log transformation, binning, linear combination, count, on numerical attributes. Some of the functions which are used in the framework are described below:

a) Confidence Interval Formulas

Confidence interval (CI) is a statistic estimation formula that uses the normal distribution for observing a point estimate by calculating maximum, minimum, median, and mean.

b) Standard Deviation

Standard deviation is a method of scaling the values based on z-score which calculates the following equation:

$$Standard\ Deviation = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{N-1}} \quad (1)$$

Where:

X_i = Value of each data point

$\bar{X}$ = Mean

N = Number

c) Logarithm Transformation Formula

Log transformation is one of the popular transformation methods used to cope with skewed data because it can remove skewness adapting the formula below.

$$x_i' = \log(x_i) \quad (2)$$

$$x_i' = \log(x_i + 1) \quad \text{in case value can be zero} \quad (3)$$

$$x_i' = \text{sgn}(x_i) log|x_i| = \frac{x^i}{|x_i|} log|x_i|$$

in case value can be negative (4)

$$x_i' = \log(x_i + \sqrt{x_i^2 + \lambda}) \quad (5)$$

generalized log transformation

d) (Linear) Regression Function

This function adapts the concept of linear or multiple regression which classifies the data by fitting two or more attributes to determine the best line. Applying regression helps to discover a mathematical equation for adjusting the data and smoothing out the noise (Fig. 3).

The equation: Let A1, ……, An be n matrices having dimension K x L.

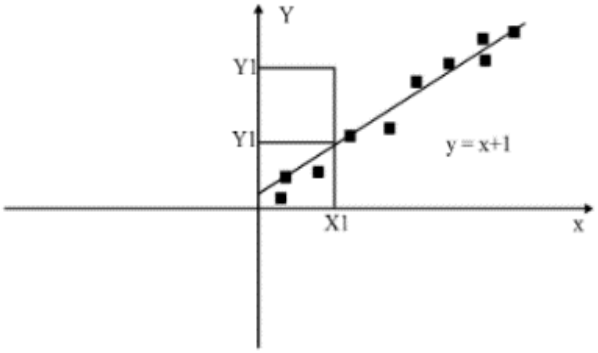

Fig. 3. Linear Regression Function.

$$B = \alpha_1 A_1 + \ldots + \alpha_n A_n \quad (6)$$

e) Clustering (K-Means)

The clustering is to group a set of spots into clusters based on a measured distance. Fraud will be recognised by locating spots out from similar clustering (see Fig. 4). All customers are classified into groups based on similar data patterns by using the K-means clustering method. K-means is an unsupervised learning algorithm that discovers the k number of clusters in a dataset. The K number of clusters are grouped by similarities based on a point at the centre of a cluster. All data are assigned to the closest cluster.

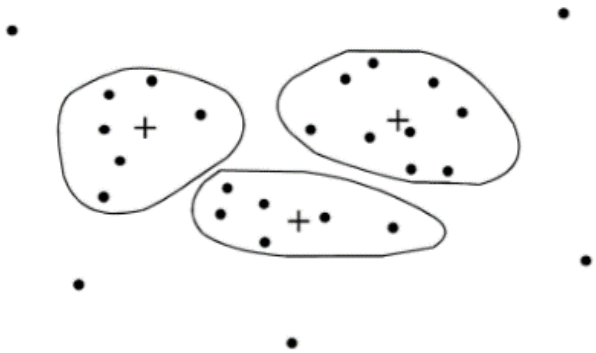

Fig. 4. Clustering.

f) Principal Component Analysis (PCA)

Principal component analysis (PCA) is an unsupervised method to reduce feature dimensions from the original feature dimensions but keeps the meaningful variation in the original attributions. PCA explores correlations among the given data and produces new aggregate variables which is a condensed dimensional feature, called principal components (PC).

In Fig. 5, the left side (plot A) shows the original data on the x-axis and y-axis. On the right-side (plot B), the 1st principal axis in the PC1 pivot displays the largest norms. PC2 pivot shows the 2nd principal axis and is orthogonal toward the pivot of PC1. The data in 2-dimension may be diminished to one dimension with extruding each element on the PC1.

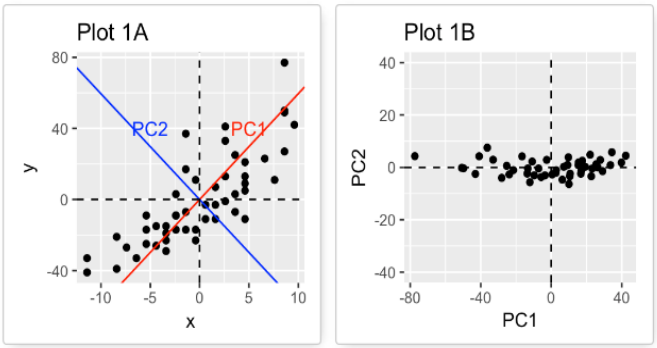

Fig. 5. Principal Component Analysis.

The mathematical approach of PCA is to maximize variances by converting a sequence of values as it is expressed in the following formula. Samples, $X_1, X_2, \ldots, X_N \in R_n$ of the variable X $\in R_n$ that was randomly selected.

$$V_{xx} = \frac{1}{N}\sum_{i=1}^{N}(X^i - \frac{1}{N}\sum_{j=1}^{N}X^j)(X^i - \frac{1}{N}\sum_{j=1}^{N}X^j)^T \quad (7)$$

$$\max_{|a|=1}\frac{1}{N}\sum_{i=1}^{N}(a^T(X^i - \frac{1}{N}\sum_j X^j))^2 = \max_{|a|=1} a^T V_{xx^a} \quad (8)$$

Where a is eigenvector corresponding to the maximum eigenvalue of a variance-covariance matrix of VX_iX_j.

B. Feature Selection Processes

In the feature creation component work introduced above, we created many additional new features. However, redundant features that correlate strongly with other features might be also included. Increasing high dimensional feature space impacts the model performance and causes overfitting, according to Mwadulo [13]. In the feature selection component, there are two main parts for selecting appropriate features from all features having both newly created features and the original data. The first part is feature measurement. In the feature measurement part, we calculate the correlation coefficient and measure feature importance, and then drop redundant features.

- Pearson Correlation Coefficient

When there are high correlations between two or more explanatory variables in the dataset, multicollinearity exists and will cause overfitting in a multiple regression model. The correlation coefficient is a statistical method to measure the degree of intensity of the relationship between feature variables. In the framework, Pearson correlation is selected to calculate the strength between two variables from different types of correlation coefficients. The range of the strength values of the correlation is expressed between -1 and 1. A value of -1 indicates the perfect negative relationship between the two feature values. On the contrary, a value of 1 indicates the perfect positive relationship between the two feature values. Values close to zero means weak or no relationship between the two values (Table II). The equation of the Pearson correlation coefficient is shown below:

$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y} \quad (9)$$

Where:

ρ_{xy} = Peason product-moment correlation coefficient

Cov (x, y) = covariance of variables x and y

$\alpha_x = standard\ dviation\ of\ x$

$\alpha_y = standard\ deviation\ of\ y$

TABLE II. BENCHMARK OF CORRELATION COEFFICIENT

Range of Correlation	Interpretation
±0.9 to ± 1.0	Very high positive (negative) corrlation
±0.7 to ± 0.9	High positive (negative) correlation
±0.5 to ± 0.7	Moderate positive (negative) correlation
±0.3 to ± 0.5	Low positive (negative) correlation
0.0 to ± 0.3	No correlation

- Feature Importance Measurement

As an evaluation method of relevant features, we select feature importance to measure the relative importance of each input feature. Scores are calculated by finding a rate of contribution indicating which features influence predictions. In a decision tree model, every node indicates a status of how to split values in an individual feature. The status depends on Gini impurity or information gain in the case of classification. While building a decision tree model, feature importance computes how much a single attribute contributes to reducing the weighted impurity.

IV. DEEP LEARNING ALGORITHM FOR FRAUD DETECTION

- Autoencoder

Autoencoders are unsupervised learning neural networks that learn to encode input data to specific features by reducing dimensions and discovering how the features can be reconstructed and decoded to the original data. In order to measure how well the input data can be reconstructed, a loss function is calculated for updating different weights and reducing the loss between the represented data and the original data. Autoencoder uses unlabeled training data {x (1), x (2), x (3), …}, where x (i) $\in$ Rn and applies backpropagation to learn how to approximate to a function h w, b (x) $\approx$ x displayed in Fig. 6. The output x^ is similar to x.

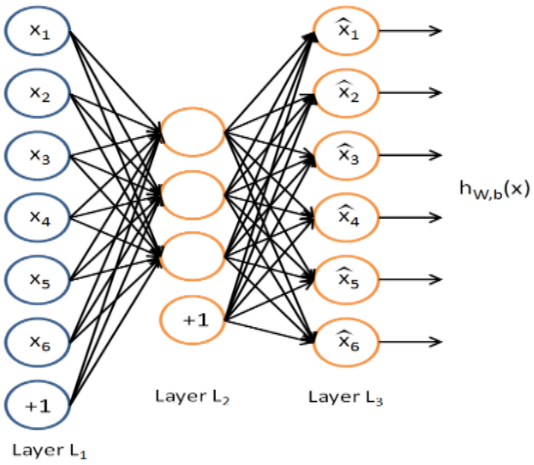

Fig. 6. Autoencoder.

There are three main layers of autoencoder: encoder, hidden and decode.

a) Encoder Layer

An autoencoder model learns how to reduce dimensions of input features and compress the given data into an encoded representation.

b) Hidden Layer

This layer holds the compressed representation of the given data and expresses the most compacted dimensional features.

c) Decoder Layer

The model learns how to reconstruct the compressed data to the original data by using the loss function and calculates the loss between the original data and the reconstructed data. The Mean square error is utilized to measure the error value shown below:

$$l(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} (xi - x^i)\hat{}2 \quad (10)$$

The equation of encoder and decoder are given as follows:

Encoder $h(x) = g((\alpha x))$

$$= \sum (Wx) or \ \tan h(Wx) \quad (11)$$

Decoder $\hat{x} = O(\alpha\hat{}(x))$

$$= \sum (W * h(x)) or \ \tan h(W * h(x)) \quad (12)$$

V. Online Banking Transaction Dataset

The online banking dataset is provided by a European bank for only academic purposes. The dataset contains about 130,000 transactions that occurred via online banking with each customer party ID. The dataset includes fraudulent actions which account for 5% of all transaction records and it is unbalanced labelled data (see Fig. 7).

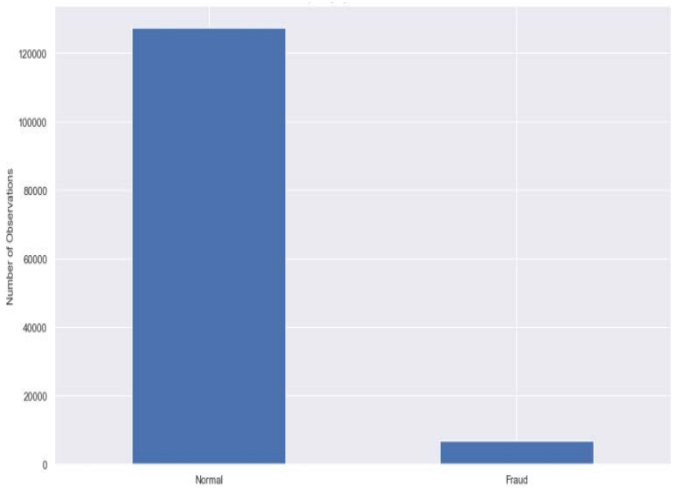

Fig. 7. An Unbalanced Target Data.

There are 39 attributes collected from various data sources such as customer information, bank account, online banking information, device information, network information, timestamp, as described in Table III. Before applying these attributes to the framework, fraudulent tendency from the point of view of transaction amount and timestamp is checked.

In the Fig. 8 shows two distributions of transaction amount frequency. One distribution (in green color) is fraudulent transactions whereas another one (in red) describes normal transactions. Both distributions show little difference between fraud and customer in this context only.

We used a logarithm function on this amount attribute, which is one of the popular mathematical functions. The histogram of log transformation is shown in Fig. 9. The distribution in green shows normal transactions while the one in red represents fraudulent transactions. From the diagnose in Fig. 8, it is shown that the fraudster in red does not steal large money at one time and seems not to be different from normal customers' transactions. It indicates that it is difficult to detect fraud transactions by the rule-based fraud detection system.

The timestamp is considered as an important feature to discover different behavior between a customer and a fraudster as customers will have their usual lifestyle patterns on a time-series basis. Days, Hours and Minutes plot transactions are shown in Fig. 10, 11, 12.

The timestamp in this dataset does not have a remarkable difference between fraud and non-fraud at a glance. Through our framework, this timestamp is segmentalized based on each customer by aggregating customer's information such as amount, network information and access information and creating new features which reveal latent customer behavior or fraudulent pattern.

TABLE III. DESCRIPTION OF ATTRIBUTES IN ORIGINAL DATA

Attribute Name	Description
ED_EVENTTYPETX	Type of event e.g., Customer Login, Make Payment etc
ED_TXNID	Transaction ID
ED_CHANNELIDENTIFIER	A way that customers can interact with a bank. This can be via the telephone, internet banking, branch, mobile.
ED_FINANCIALINSTITUTENM	Financial Institute name
ED_SUBCHANNELNM	Sub-channel name
CUSTD_PARTYID	Customer Party ID
CUSTD_EMAILADDRESSTX	Customer's email address
EVENT	Event of transaction
AUTO_RESPONSE	Auto-response
LATENCY	Latency
IDVD_LOGINTYPE	Login Type
ACTD_BANKACCTNO	Account's bank account number
ACTD_ACCTTYPENM	Account type
ACTD_AVAILABLEBL	Available balance
TRNSD_BENEFICIARYSORTCD	Beneficiary sort code
TRNSD_BENEFICIARYACCTNO	Beneficiary account number
TRNSD_TRNSAM	Transaction amount
TRNSD_PAYMENTDT	Transaction Datetime
TRNSD_TXNREFERENCETX	Transaction reference
TRNSD_PAYMENTDT	Transaction Date Time
IDVD_AUTHENTICD	Authentication code
IDVD_INTESESSIONID	Internet session ID
IDVD_IPADDRESSID	IP address
IDVD_CLIENTSCREENRESOID	Client screen resolution
IDVD_USERAGENTTX	User-agent
IDVD_DEVICEID	Device ID
IDVD_INTESESSIONID	Internet session ID
IDVD_CLIENTSCREENRESOID	Client screen resolution
IDVD_USERAGENTTX	User-agent
IDVD_BROWSERLANGTX	Browser language
IDVD_IPADDRESSID	IP address
IDVD_DEVICEID	Device ID
IDVD_TELSESSIONID	Telephone session ID
IDVDATA_TRNSTS	Transactions timestamps
EVENT	Event of transaction
AUTO_RESPONSE	Auto-response
Last_LATENCY	Latency
IDVD_LOGINTYPE	Login Type
IDVD_AUTHDETAILS1	Authentication details
Is Fraud	Fraud flag whether fraud or not

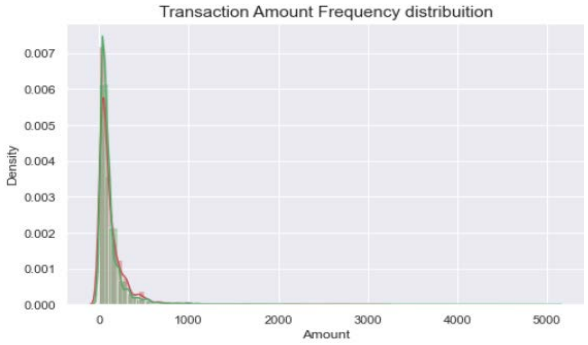

Fig. 8. Fraudulent and Customer's Distributions of Transaction Amount.

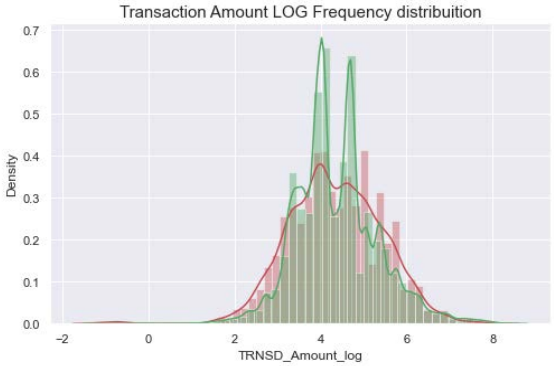

Fig. 9. Distributions of Transaction Amount with Log Transformation.

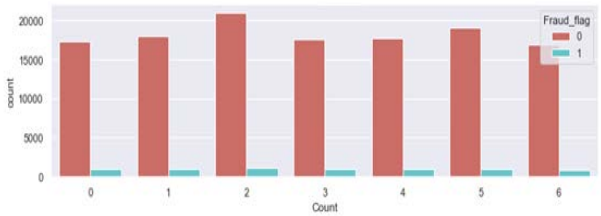

Fig. 10. Transactions base on Weekdays.

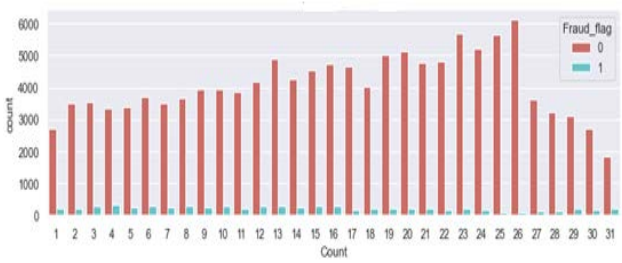

Fig. 11. Transactions base on Days.

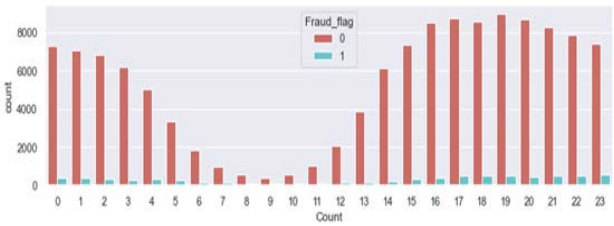

Fig. 12. Transactions base on Hours.

VI. Experiments and Results

A. Experiments

From our previous published work in [19], it has been demonstrated that the performance of fraud detection models with prepared feature sets, performs better than the models with original data only.

The purpose of the experiments in this research is to verify the effectiveness of using a feature set that is created through the processes using our framework for deep learning. The online banking dataset described in Section 3 is used. Target attributes in original data that are used for implementation of feature aggregation and transformation methods are almost fixed specifically for online banking transaction data. This is because the banking system has common attributes in some tables such as customer information, banking information, network information. According to processes in the feature creation component, we apply feature engineering methods on the original data and create 65 new features after aggregating features and transforming features (see Table IV).

TABLE IV. Created New Features

New Attributes Created by Feature Engineering
LATUPDATE_Weekdays
LATUPDATE_Hours
LATUPDATE_Days
LATUPDATE_Minute
BALANCE_log
Balance_min_mean
Balance_min_std
Amount_log
AUTO_RESPONSE_std
ACTD_AVAILABLEBALANCE_log
ACTD_AVAILABLEBALANCE_min_mean
ACTD_AVAILABLEBALANCE_min_std
Trans_min_mean
Trans_min_std
mean_last
mean_last_count
mean_balance
min_last
min_balance
min_last_balance
max_last
max_balance
max_last_baalance
count_last
count_balance
count_last_balane
mean_last
mean_balance
mean_last_balance
max_last
max_balance
max_last_balance
CUSTD_PARTYID_IDVD_CLIENTSCREENRESOID_count_LATUPDATE_Weekdays
CUSTD_PARTYID_IPADDRESSID_count_LATUPDATE_Weekdays
CUSTD_PARTYID_IDVDATE_TRNSTS_count_LATUPDATE_Weekdays
CUSTD_PARTYID_LAST_LATENCY_count_LATUPDATE_Weekdays
CUSTD_PARTYID_TRNSD_TRANSSESSIONCD_count_LATUPDATE_Weekdays
CUSTD_PARTYID_TRNSD_Amonut_log_count_LATUPDATE_Weekdays
CUSTD_PARTYID_ACTD_AVAILABLBALANCE_log_count_LATUPDATE_Weekday
CUSTD_PARTYID_ACCESS_CD_count_LATUPDATE_Weekdays
CUSTD_PARTYID_TRNSD_Amount_log_count_LATUPDATE_weekdays
CUSTD_PARTYID_TRNSD_Amount_count_LATUPDATE_Weekdays
LATENCY1_std
LATENCY2_std
LAST_LATENCY_std
LGIN_LATENCY1
LGIN_LATENCY2
LATENCY1_std
LATENCY2_std
clusters_1
clusters_2
clusters_3
count_cluser
Days_std
Weekday_std
Hours_std
PCA_EVENT0
PCA_EVENT1
PCA_PASS0
PCA_PASS1
PCA_PASS2
PCA_FinancialInfo0
PCA_FinancialInfo1
PCA_FinancialInfo2
PCA_CustomerID_IP_Amount

The result of feature importance measurement is presented in Table V. We measured the feature importance of all features both original and the created features and recognized that the most of features with higher importance rate are the new features created via the feature engineering framework. Based on higher scores, we selected 57 features among all 104 features and the rest of feature's importance rate were nearly equal to zero.

TABLE V. FEATURE IMPORTANCE MEASUREMENT (TOP30)

Attribute	Importance
count_last_balance	0.103799286
CUSTD_PARTYID_ACCESS_CD_count _LATUPDATE_Weekdays	0.095292695
min_last_balance	0.094270386
count_balance	0.091391504
count_last	0.07332497
TRNSD_BENEFICIARYACCTNO	0.064115062
BALANCE_log	0.060077297
Balance_min_mean	0.05803433
IDVDATA_TRNSTS	0.04700098
CUSTD_PARTYID	0.041534992
CUSTD_PARTYID_TRNSD_Amount_log_count _LATUPDATE_Weekdays	0.041354892
mean_last_balance	0.025604612
mean_balance	0.024146388
min_last	0.015255225
ACTD_AVAILABLEBLBALANCE_min_mean	0.014513752
ACTD_AVAILABLEBLBALANCE_log	0.013771143
IDVD_SCREENSIZE	0.013566704
TRNSD_TRANSSESSIONCD	0.011465614
LATENCY1_std	0.010989958
PCAID_D0	0.010124042
CUSTD_PARTYID_ACTD_AVAILABLEBLBALANCE _log_count_LATUPDATE_Weekdays	0.009920134
LGIN_LATENCY1	0.009015999
max_last_balance	0.008283598
ACTD_AVAILABLEBLBALANCE_min_std	0.007470134
CUSTD_PARTYID_TRNSD_TRANSSESSIONCD _count_LATUPDATE_Weekdays	0.004289627
PCAID_D1	0.003624484
PCA_PASS0	0.003490054
LATUPDATE_Days	0.003359021
max_last	0.00317593
Days_std	0.003111183

Now, three different deep learning models are built with three types of feature sets: (1) original dataset only, (2) original dataset plus newly created features, (3) only selected features following feature importance scores (see Tables VI to VIII).

We then use Tensor Flow which provides a simple autoencoder program from Python libraries.

Autoencoder requires the setting of some parameters and we manually determine optimal parameter values. The Autoencoder algorithm is applied in the settings below:

1) The data are divided into 80% training data and 20% testing data. The training data consists of customer transactions only excluding fraudulent data. In the testing, the autoencoder encodes and compresses the input data and tries to represent the original data based on leaned dimensional reduction and reconstruction. Then, it can distinguish a fraudulent transaction if it cannot represent the data again.

2) The number and size of layers are set from left to right 57-18-10-6-6-10-18-57 in the case of the selected feature set. These numbers show how to encode and decode in the neural networks. From the fifth to the eighth layers the data is reconstructed, and the mean squared error as a loss function is calculated. The significant point in the layers is that the number of input data size is the same as the output data size.

TABLE VI. PARAMETERS OF AUTOENCODER

Parameter Name	Value
Optimizer	Adam Optimize
Loss Function	Mean_Squared_Error
# of Epoc	1000
Batch Size	128
Test_size	0.2

TABLE VII. AUTOENCODER MODEL USING TENSORFLOW

```
input_layer = Input (shape= (input_dim,))
encoder = Dense (encoding_dim, activation=" tanh", activity_regularizer =
regularizers. l1(learning_rate)) (input_layer)
encoder = Dense (hidden_dim1, activation =" elu") (encoder)
encoder = Dense (hidden_dim2, activation =" tanh") (encoder)
decoder = Dense (hidden_dim2, activation = "elu") (encoder)
decoder = Dense (hidden_dim1, activation = "tanh") (decoder)
decoder = Dense (input_dim, activation =" elu") (decoder)
autoencoder = Mode (inputs = input_layer, outputs = decoder)
```

TABLE VIII. AUTOENCODER LAYERS (SELECTED FEATURES)

Layer (type)	Output Shape	Param #
input_1 (Input Layer)	[(None, 57)]	0
dense_(Dense)	(None, 18)	1044
dense_1 (Dense)	(None, 10)	190
dense_2 (Dense)	(None, 6)	66
dense_3 (Dense)	(None, 6)	42
dense_4 (Dense)	(None, 10)	70
dense_5 (Dense)	(None, 57)	627
Total params: 2,039 Trainable params: 2,039		627

B. Performance Metrics for Fraud Detection Models

Classification problems using unbalanced labelled data cannot be evaluated by the accuracy only. Especially in the case of financial fraud detection, we should evaluate and compare the model performance with plural metrics because the classification problem is necessary to be considered as a balance between the true positives ratio (TP) and the false-positive ratio (FP). TP is the number of predictions as fraud where the actual result is also fraud. FP is the number of predictions as a legitimate transaction where the actual result is the customer. The true negatives (TN) and the false negatives (FN) are also significant metrics when measuring the performance of recall and precision. A recall is the ratio of frauds that are perfectly classified whereas precision is the ratio of the accuracy of fraud predictions. When the score of recall is high, it indicates a poor rate of FN which is the number of predictions as a legitimate transaction where the actual result is fraud. When the score of precision is high, it indicates a poor rate of FP which is the number of predictions as fraud where the actual result is the customer.

F1-Measure is the harmonic mean of precision and recall. The best score is 1 whereas the worst score is 0. This metric seeks the balance between precision and recall (see Table IX).

TABLE IX. PERFORMANCE METRICS DEFINITION

Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F1-measure	2*Precision*Recall/ (Precision + Recall)

The confusion matrix shows a matrix describing the performance of the model using True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) (see Table X).

TABLE X. CONFUSION MATRIX

# of Observations	**Predicted Normal**	**Predicted Fraud**
Actual Normal	TN	FP
Actual Fraud	FN	TP

C. Results and Evaluations

The effectiveness of the feature engineering framework is measured by comparison with the performance of the autoencoder model with the original data only. As stated in the previous section, in order to evaluate the efficiency of the created and selected features, the model performance is assessed by AUC, recall, precision, and F-measure. The following Table XI shows the comparison of autoencoder models with two threshold values (Threshold=4 and 1) with three different types of datasets.

TABLE XI. COMPARISON OF AUTOENCODER MODELS WITH THRESHOLD VALUE =4 IN THE DIFFERENT DATASETS

Threshold=4	**AUC**	**Recall**	**Precision**	**F-measure**
Model1 with only original data	0.65	0.058	0.188	0.0896
Model 2 with original data plus new features	0.83	0.064	0.215	0.0986
Model 3 with the selected features	0.92	0.064	0.215	0.0986

The results in Table XI shows that model 3 with the selected features and model 2 with original data plus newly created features are higher in all performance metrics than model 1 (with original data only). Model 3 has a higher AUC than Model 2.

Different threshold values are chosen based on the situation shown in Fig. 13. We can adjust the threshold value to a better classification part.

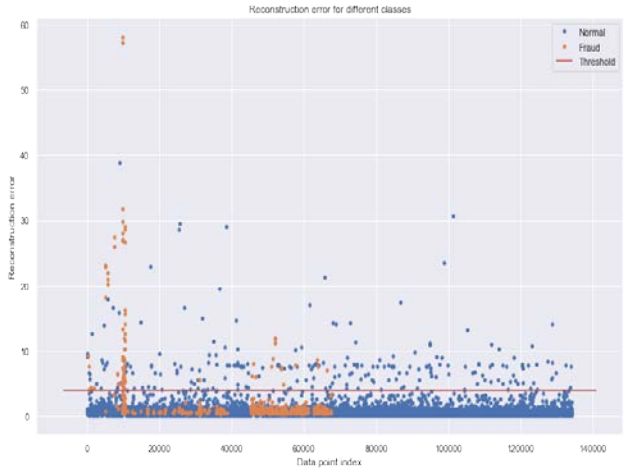

Fig. 13. Data Distribution in Threshold 4.

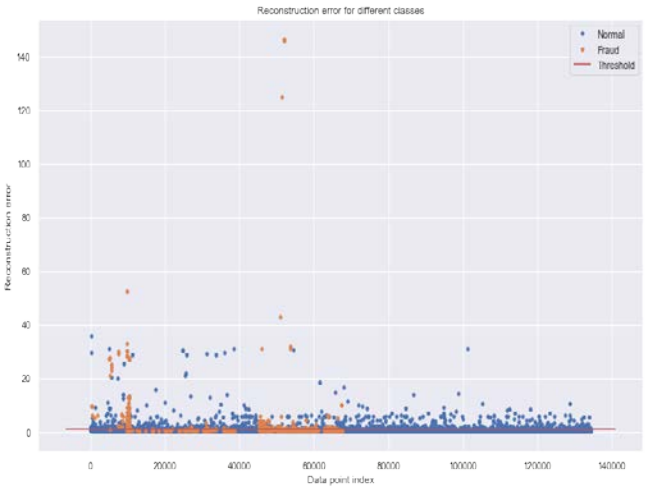

Fig. 14. Data Distribution in Threshold 1.

The data distribution presents the thresholds between fraudulent data and legitimate data. In the case of threshold 4, most fraud transactions are not classified well with the confusion matrix confirming this. Now let's change the threshold value from 4 to 1. Fig. 14 shows the data distribution when the threshold value is equal to 1.

It appears that fraud transactions with the threshold value of 1 are better classified than the ones with a threshold equal to 4. The performance of the models with a threshold value of 1 becomes as described in the table below:

TABLE XII. COMPARISON OF AUTOENCODER MODELS WITH THRESHOLD VALUE=1 IN THE DIFFERENT DATASETS

Threshold=1	AUC	Recall	Precision	F-measure
Model1 with only original data	0.73	0.161	0.104	0.1263
Model 2 with original data plus new features	0.91	0.358	0.451	0.3994
Model 3 with the selected features	0.96	0.648	0.430	0.5167

Table XII demonstrates the superiority of Model 3 with selected features.

From the above results two suggestions are drawn. First, the performance of the autoencoder models significantly improves when the new features are created based on the proposed feature engineering framework. The experiments indicate that it is efficient for a deep learning model (for classification) to implement feature engineering on original data before inputting the data. Second, adjustment of threshold in autoencoder also made an impact on the model accuracy. A combination of an appropriate setting of threshold and optimal feature set can improve a deep learning model performance for fraud classification.

Another point of view from the experiments is about scores of Recall and Precision of each model. Recall has an impact on huge money loss whereas Precision influences customer satisfaction and confidence. All measurements of the models using the selected feature set are the highest scores than model 1. Precision in model 2 is higher than the precision in model 3 which indicates that a balance between precision and recall is a trade-off. As mentioned above, a high score of recall indicates the model can identify fraudulent activities without mislabeling them as actual customers. On the other hand, a high score of precision means the model can identify actual customers' transactions without mislabeling them as fraud. In either case, using the created features in the newly built feature engineering framework could improve the model performance and detect fraudulent transactions based on the reconstruction of the input data.

VII. CONCLUSION

A fraud detection system in recent years adopts machine learning models which learn anomaly data patterns from past transaction records. However, the total losses through online banking in the United Kingdom have been increasing because fraud schemes continue to further evolve as the online payment system advances. Feature engineering is a key to improving the accuracy of fraud detection models and can reveal latent data patterns by transforming raw data into another dimension. In the paper, we used the feature engineering framework which creates new features and selects effective features through feature engineering techniques for autoencoder, a deep neural network, this time. As a result, the performance of the autoencoder models built with selected features from the framework was better in comparison to the performance of the autoencoder models built with raw data only.

Deep learning methods have a function of feature extraction to reduce the number of features in an input data and automatically learns features at multiple levels by combining the input features. Although they already have a part of feature engineering function in the algorithms, using the prepared dataset including new features created through the feature engineering framework was more effective for improving the deep learning model performance.

In this paper, we used an autoencoder as a deep learning model and presented the effectiveness of using the feature engineering framework. In further work, we will use other deep learning algorithms such as recurrent neural network (RNN) and convolutional neural network (CNN) which are often used for financial fraud detection. Moreover, the feature selection component in the framework will be studied and improved.

REFERENCES

[1] FRAUD – THE FACTS 2021:The definitive overview of payment industry fraud:UK Finance, 2021.

[2] Casilda Aresti, "Technology and operations management: PayPal's Use of Machine Learning to Enhance Fraud Detection" (PayPal's Use of Machine Learning to Enhance Fraud Detection (and more) - Technology and Operations Management (hbs.edu)) , 2018.

[3] Niccolo Mejia, "AI-Based Fraud Detection in Banking – Current Applications and Trends" (AI-Based Fraud Detection in Banking – Current Applications and Trends | Emerj), 2020.

[4] J.Jordan. "Introduction to autoencoders" [Cited:14/04/2021] https://www.jeremyjordan.me/autoencoders/. 2018.

[5] P. Jiang, J. Zhang and J Zou. "Credit Card Fraud Detection Using Autoencoder Neural Network", Cornell University, arXiv.org. vol. 1, site:1908.11553, 2021.

[6] M.A. AI-Shabi. "Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Dataset", ELSEVIER, Expert Systems With Applications vol.167, pp.254–262, doi: 10.1016/j.procs.2020.03.219, 2019.

[7] S.Misra, S.Thakur, M.Ghosh, K.Saha. An Autoencoder Based Model for Detecting Fraudulent Credit Card Transaction 2020.

[8] A. Pumsirirat and L.Yan. "Credit CardFraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine", International Journal of Advanced Computer Science and Applications(IJACSA), Volume 9 Issue 1, 2018.

[9] Y. Lucas, P.Portier and S.Calabretto. "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs", ELSEVIER, Expert Systems With Applications vol.102, pp.393–402, doi: 10.1016/j.future.2019.08.029, 2020.

[10] X.Zhang, Y.Han, W.Xu and Q.Wang. "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture", ELSEVIER, Expert Systems With Applications vol.557, pp.302–316,doi: 10.1016/j.ins.2019.05.023, 2021.

[11] J.S. Kalwihura and R. Logeswaran. "Auto-Insurance fraud detection: a behavioural feature engineering approach",Cornell University, arXiv.org. site: 1805/1805.09741.pdf, 2020.

[12] A.C. Bahnsen, D.Aouada and S.B. Ottersterm. “Feature engineering for credit card fraud detection”, ELSEVIER, Expert Systems With Applications vol.51, pp.134–142, 2016,doi: 10.1016/j.eswa.2015.12.030.

[13] M.Mwadulo. “A review on feature selection methods for classification tasks”, 2016, International Journal of Computer Applications Technology and Research, Vol. 5, Issue 6, pp. 395-402, 2016, ISSN:-2319–8656.

[14] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty and V. Sravan Kiran, "Similarity Based Feature Transformation for Network Anomaly Detection," in IEEE Access, vol. 8, pp. 39184-39196, 2020, doi: 10.1109/ACCESS.2020.2975716.

[15] R.Wedgy, J.Max.Kanter, K.Veeramachaneni, S.M.Rubio and S.l.Perez. “Solving the false positive problem in fraud prediction using animated feature engineering.”, Cornell University, arXiv.org, Computer Science, doi: 10.1007/978-3-030-10997-4_23.

[16] P.Jiang, J.Zhang and J.Zou. ”Credit card fraud detection for autoencoder neural network.”, Cornell University, arXiv.org. site:1908.11553v1,2018.

[17] A.Pumsiriat and L.Yan. “Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine.”, 2018, the International Journal of Advanced Computer Science and Applications(IJACSA),vol. 9, doi:10.14569/IJACSA.2018.090103.

[18] M.A.Al-Shabi. “Credit card fraud detection using auto encoder model in unbalanced datasets.”, 2019, Journal of Advances in Mathematics and Computer Science, pp. 1-16, doi: 10.9734/jamcs/2019/v33i530192.

[19] C.Ikeda K.Ouazzane and Q.Yu, “A new framework of feature engineering for machine learning in financial fraud detection.”, 2020 10th International Conference on Artificial Intelligence, Soft Computing and Applications, London, 2021 vol 10, doi:10.5121/csit.2020.101517.