NTNU
Norwegian University of
Science and Technology

Deep Learning Based Approaches for Financial Fraud Detection

Nan Zhang

15-12-2020

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Information Security and Communication Technology
Norwegian University of Science and Technology, 2021

Supervisor: Prof. Sule Yildirim Yayilgan

Preface

This thesis is written in autumn 2020 which concludes several years of learning at NTNU. It is inspired by the popularity of deep learning techniques and the need for financial risk control of cradit card issuers. The intended audience of this thesis are security practitioners and enthusiasts who are interested in financial risk control.

15-12-2020

Acknowledgment

As an international students, I spend a lot of years in Norway. When I came to Norway, I was just a little girl who was not mature to face that much challenges in life, and at that time I had a lot of fancy about life. With time past, I made many mistakes and left many regrets in my life. But I also feel lucky that I still alive and I haven't loss confidence in life. During this process, I want to acknowledge Hilde Bakke who encouraged me many times and gave me many chances to restart. Besides, I also want to acknowledge my professor Sule Yildirim Yayilgan who have accepted me to follow her to do my master thesis as a distance student. Further more, I want to acknowledge my family who connive me to be a capricious girl for so many years. And know, I want to say that I am an independent woman who will be responsible for the rest of my life.

N.Z.

Abstract

Financial fraud detection is an annoying problem which takes financial institutions a lot of money and energy to reduce the loss caused by it. Traditional fraud detection methods need a lot of trained audits to verify business cases which is pretty inefficient, with the prevalent of online payment there is a strong need for automated fraud detection system. It should be able to detect fraud from large amount of transaction data in short time without intercepting too much normal behaviors. As fraud detection in business institution is not a new topic, there are already many solutions to this problem such as imbalance learning for dataset imbalance problem, GBDT for fraud detection. Recently, with the development of deep learning techniques, there are many attempts which try to use deep neuron networks for financial fraud detection. However none of them conduct a comprehensive analysis on this field. This thesis introduces a complete fraud detection methodology which tries to utilize deep neuron networks to solve problems existed in the entire process of financial fraud detection. According to our experiment results, we got three conclusions: (1) deep neuron networks can simplify the process of feature engineering. The proposed CNN and LSTM networks have obtained better prediction accuracy with underived feature set than LightGBM(a widely used model in financial fraud detection field) with a complete feature set. (2) Deep learning based oversampling method(Autoencoder) can alleviate the influence of dataset imbalance, the performance of it is similar to the classical oversampling method SMOTE. (3) Proposed deep neuron networks (CNN and LSTM) perform much better than base model (LightGBM) with the same dataset, this verify the hypothesis that deep neuron network is a powerful tool which can improve the efficiency of fraud detection. (4) Two dataset are used for testing the three classifiers used in this thesis, and the experiment results on these two dataset are similar which further confirm the conclusion we have obtained above.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Topic covered by the project

Financial fraud is a problem which influences normal business activities and draws a lot of attention from business organizations. It can be defined as using illegal approaches to obtain economic benefit. With the widely use of internet, online finance is a new trend bringing opportunity as well as challenge. It can find more potential customers, produce diverse business products and simplify the complex procedure of traditional financial product. Every advancement has two sides, the low entrance stander pose high dangerous to the business organizations. Some people deliberately borrow money without plans to return cause a big loss to these organizations. Thus, better risk management approaches are needed.

For many years, large organizations such as bank hire a large amount of people to audit cases which has potential risk of fraud. However, this cost a lot of money and the auditing process is pretty slow. With the advancement of technology, artificial intelligence and big data techniques bring a revolution to every field including financial fraud detection. Fraudsters are a group of smart people which try to find loophole in rules and benefit from it. Various fraud methods challenge the financial employers to evolve their ability of fraud detection. Machine learning and statistical methods have already been proven to be useful for financial fraud detection such as credit card fraud detection, stock price predicting. Fraud detection is a two class classification problem with imbalanced dataset. There are already some popular techniques and procedures in industry for fraud detection such as xgboost, logistic regression. Nevertheless, new approaches are needed to be researched to combat the guileful fraudsters.

Deep learning is the hotest research field in recent year which has obtained great success in image processing and nature language processing. The deep network structure is a powerful tool to extract complex information from unstructure data. Thus, many researchers try to mitigate deep network to financial fraud detection area. Aleskerov et.[7] propose a one layer neural network for credit card detection as early as 1997. However, due to the limitation of deep learning technique and the special structure financial data, it has not been widely used in industry in recent year. Thus, we believe that it deserve to conduct a research on applying deep learning algorithms on financial fraud detection and find the possible to utilize this technique in this area.

1.2 Keywords

Financial Fraud Detection, Deep Learning, Machine Learning, Feature Engineering

1.3 Problem description

With the revolution of digitalization, online finance generates many opportunities to financial organizations as well as fraudsters. According to the study of Association of Certified Fraud Examiners, fraud results in 5% revenue lost every year for every organization[8]. It is obviously that reducing fraud rate is an effective way for increasing financial organization incomes. However, the big data era put forward more challeges to financial organizations for fraud detection due to the volume, variety and velocity feature of it. Besides, as financial data has its special features such as structured, high-dimensional and imbalance distribution, new approaches for financial fraud detection are needed.

There are some material methods which have been widely accepted in industries such as logistic regression and random forest. However, compared with other field, technique progress goes pretty slow in financial fraud detection. The main reason is that financial industry is a field which pay more attention to security and privacy than technique. Thus, employers prefer material methods which can be understand easily and have relatively good performance.

In recent year, the breakthrough of deep learning research cause a revolution in computater vision and nature language processing. As the inner operation of neural network is hard to explain, financial organizations are reluctant to take this method in industry use. But, due to the power of deep neural network, we think it desires our work to conduct the research on applying deep learning algorithms on financial fraud detection.

1.4 Justification, motivation and benefits

The idea of this topic is come from the popularity of deep learning methods. I have experience in applying deep learning algorithms in image processing such as face detection. The power of deep neural network leaves a deep impression on me. And I though it may get good result in other area.

After getting offer from a state-owned bank, I thought it may be a good attempt to apply deep learning algorithms on financial data. This gives me an opportunity to familiar with business procedure and mitigate the knowledge of deep learning to other field.

Financial fraud is an inevitable dangerous factor for every organization. Traditional financial fraud includes: bank fraud, corporate fraud and insurance fraud. With the generating of online business, the fraud type becomes varied. Deep learning based fraud detection method is a powerful tool to find anomalous bahavior in massive data. The purpose of this thesis is offering a solution to fraud detection and we think this method is also useful for other structure data like electricity data.

1.5 Research questions

This thesis is aiming at applying deep learning algorithms in financial fraud field. Because financial fraud data has its special features, the research questions are devised for these problems:

1 How to use deep learning technique to solve the dataset imbalance problem of financial fraud data?
2 Can deep learning based classifier perform better than other machine learning methods?

3 Does deep learning based classifier simplify the process of feature engineering?

Our research questions try to use deep learning technique to solve the three common problems in financial fraud detection which will be encountered for every financial fraud detection researchers. Usually the method for each question is different, but in this thesis we will explore the potential power of deep learning in the whole process of financial fraud detection.

1.6 Contributions

The main contribution of this thesis is conducting a comprehensive analysis on financial fraud detection problem and trying to use deep learning techniques to solve problems existed in this field. We explore the potentiality of deep neuron networks on three main challenges in financial fraud detection: (1) simplifying the process of feature engineering; (2) balancing the dataset; (3) increasing the performance of classifiers. The experiment results verify that our proposed methods outperform than classical methods which are promising methods for financial fraud detection. To the best of our knowledge, we are the first to apply deep neuron networks on dealing with problems in entire process of financial fraud detection. In addition, we conduct a complete and canonical process of feature engineering which can be migrated to any other tabular data, and illustrate the results of feature engineering in experiment part.

1.7 Thesis Structure

The remainder of this thesis is structured as follows:

Chapter 2 gives a detail introduction on backgroud knowledge used in this thesis which includes: brief introduction of fraud risk, basic concept of machine learning and deep learning, general explanation on feature engineering methods and imbalanced learning methods.

Chapter 3 introduces the state-of-art researchers about using deep neuron networks to solve problems existed in financial fraud detection.

Chapter 4 explains methods selected for financial fraud detection in detail. It follows steps used for fraud detection and explains methods used in every step thoroughly.

Chapter 5 illustrates experiment results of proposed methodologies and compares the outputs with other canonical methods' results. Based on experiment results, we conduct a series of analysis.

Chapter 6 gives out the answer of research questions, concludes findings of experiments and has a quick look at future work.

2 Background

2.1 Fraud Risk

There are two type of risk in credit loan: credit risk and fraud risk. Credit risk mainly focus on assessing capability of earning money and willing to return money of the applicant. Fraud risk is judging the purpose of applicant. For financial organizations, the risk of credit risk is manageable by risk pricing and provisions. When the purpose of the applicant is cheating money and the financial organization cannot detect the fraud on time, the lost will be huge. Usually the lost caused by one fraud case need several loans to make up. If the fraud rate over a threshold, it is hard for the financial organization to make money. Recently, there is a trend that the fraud is committed by a gang and this will cause severe consequence to the financial organization. Hence, the attitude of financial organization towards business fraud is strict, some organizations may sacrifice a portion of normal applicants to reduce the fraud risk.

There is no way to prevent fraudster conducting crimes and the best way to protect financial organizations is using a complete protection life cycle at the financial organization end. The strategies of anti-fraud life cycle is devised for every stage of applicant take loan from financial organizations. The anti-fraud life cycle is as follows:

First protection level is located in equipment and network. Similar with network intrusion detection, anti-fraud life cycle is starting from hardware protection. The common devices being checked includes: proxy detection, IDC detection, simulator detection and Trojan detection.

Second protection level is user behavior detection. Abnormal behavior represents potential risk. For example, if thousands of registration happened in a short time, there is a high chance that some people or organizations try to use malware for registration. The abnormal behavior includes: registration, login, transaction, event and abnormal time interval.

Third protection level is checking business frequency. It is an important index for financial fraud. There are several import business frequencies usually being used by analyzers such as registration frequency, login frequency, transaction frequency, region frequency and time interval frequency.

Fourth protection level is observing business exception. Exception can be normal behavior which does not follow the regular routine of applicant. For example,the applicant lived in Norway suddenly takes out money from America. The business exception includes: registration exception, login exception, transaction exception, region exception and time period exception.

Fifth protection level is using knowledge graph for fraud group detection. Nowadays, many financial fraud is launched by a gang of fraudster. There is a famous car insurance fraud detected by graph based method. Two drivers exchange the identity to perform accidents in two different place for cheating insurance. The fraud case is detected by insurance company successfully through the construction of social network. The graph based method is fairly popular in crime detection

research and is taken as the last protection level of our anti-fraud life cycle.

The countermeasures for financial fraud are: making filtering rules to intercept suspicious applicants and using algorithms to detect fraudsters. In financial industry, the two methods are combined together for fraud detection. The rule based method makes the coarse-grained selection for the applicants. For those well disguised applicants, fraud detection model performs good in past decades.

Rule based method is widely used in fraud detection. The easier to use and good performance features make it a basic tool for fraud detection. The disadvantages of it are:

- The strong rule based method has a high false positive rate which may misclassify the normal applicant into the black list.
- It cannot give a concrete fraud score.
- It does not take the fraud risk shifting from credit risk into consideration, especially during the depression period.

Statistical and machine learning based models generate a fraud score which leave a space for salers to find more potential customers instead of rejecting. Models can calculate the possibility of shifting from credit risk to fraud risk and financial organizations can utilize these data for risk management.

For financial fraud detection, the most important task is how to find black samples from asymmetric distributed data. From researchers perspective, solutions to solve such problems can be coarsely classified into two categories: unsupervised learning and supervised learning methods. Due to the special characteristic of fraud detection, the distribution of dataset always be imbalanced. Fraud activity is far less than normal activity, and crafty fraudster try their best to make fraud behavior looks like normal behavior. Hence, model selection for fraud detection is based on the quality of dataset. When the quality of dataset for training is good which means every sample has a label, supervised learning methods perform well on this type dataset. Classical supervised learning methods such as logistic regression, XGBoost have been widely used in financial fraud detection. These methods not only have good performance on labeled dataset, but also suitable for tabular data especially in financial fraud scenery. However, in practical situation, there are not enough labeled training data. We need to detect fraud activities from large raw data without label. Outlier detection is a group of methods for such problems. Besides, if a small portion of samples has labels, semi-supervised learning is suitable for this situation. In some extreme situation, rules extracted from experienced expertise can be used for solving the cold start with no data problem.

2.2 Machine Learning

Machine learning is a interactive and iterative process which is used to extract useful information from massive data. It can be thought as the subset of data mining but not absolutely overlapped with it as it includes other fields such as computational learning theory. Usually, we use machine learning to solve practical problems which trying to find the hidden regular of raw data. Typical machine learning process includes two phases: training phase and classification phase. Training phase uses training dataset to generate a model for the classification task. Then classification phase

uses validation dataset to test the performance of the trained model. Once the performance of the trained model is satisfied the requirements, it can be used to solve the practical problem.

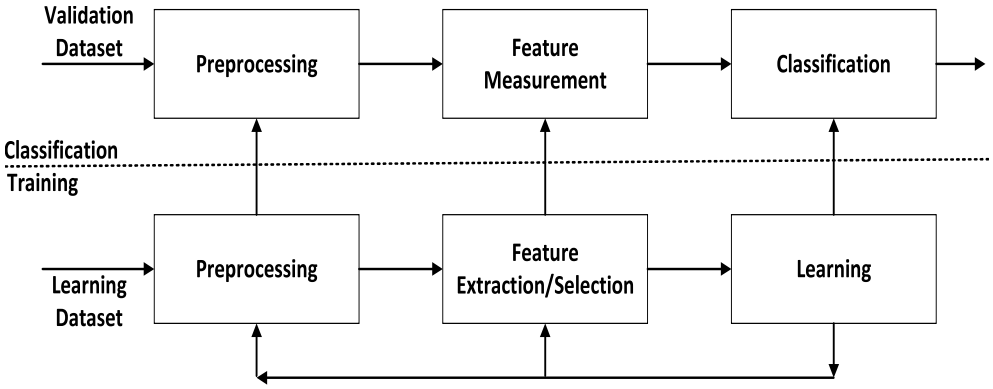


Figure 1: Machine learning process[1]

Kononenko and Kukar[1] in their book classified machine learning into three taxonomies: supervised learning, unsupervised learning and reinforcement learning. Each taxonomy includes a series of methods which have similar idea while own special feature for different practical problems. The three taxonomies can be further seperated into clustering, associations, inductive logic programming, equations, classification, regression and reinforcement learning according to how the knowledge is used.

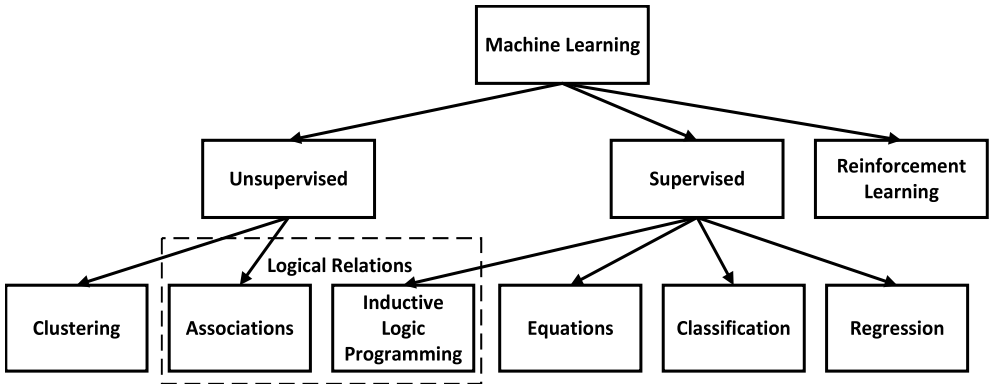


Figure 2: Machine learning taxonomy[2]

Among these methods, classification and regression methods are the most popular methods which are widely used in many fields such as disease diagnosis. Classification can be defined as finding a model $f(x) \rightarrow y$ by using a dataset D. The dataset contains n samples and each sample is composed of a feature vector x and a class label y. The dataset looks like $D = \{(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)\}$ where x is a feature set including m dimensions and y is a number which indicate the class of the

sample belongs to. Usually, the dataset D is divided into two subsets which are training set and validation set. The training set is used for training the model for classification and validation set is used for testing the performance of the obtained model. The number of classes is not limited and the most common classification problem is binary classification problem, for example, gender prediction. The purpose of training phase is finding the function which can separate the dataset into different classes. The validation phase is using validation dataset to test the performance of trained model. If the performance of the obtained model over a predefined threshold, it can be used to predict an unseen sample. Classical classification algorithms are: support vector machine(SVM), decision trees, neural networks etc. Regression methods use a real value as the target y, the desired output of the trained model is as close as possible to class label. The difference between classification and regression is the performance measurement method. Regression uses the average of difference between the predict label with correct label and classification uses the proportion of same output between predict label and correct label.

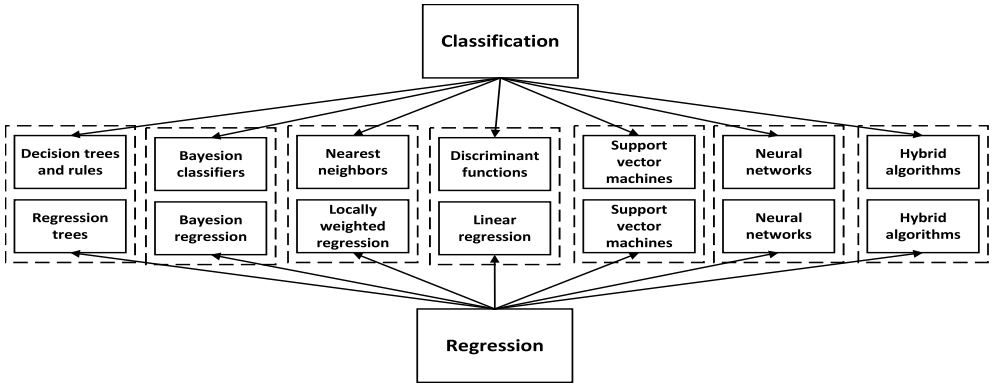


Figure 3: Classification and regression taxonomy[2]

As financial fraud detection is a two binary classification problem, the following paragraph will focus on the methods used in this thesis. Currently, the most popular machine learning methods in financial fraud detection field are: logistic regression and ensemble learning such as xgboost and lightgbm.

Ensemble Learning

The main purpose of supervised learning is finding a stable model which has good performance for the practical problem. However, sometimes we can only get the model whose performance is relatively good which cannot satisfied our requirement. Ensemble learning combines multiple week model to get a strong model which perform good for our specific problem. The hidden idea behind ensemble learning is using several model to correct the error made by one model. Ensemble learning can be thought as the combination of several machine learning techniques to reach the goal of minimizing variance(bagging), deviation(boosting) and improving predict accuracy(stacking). Ensemble learning has different strategies on different dataset: the large dataset is divided into

multiple small dataset to train different models which are used to form the final model, the small dataset will be sampled many times to get multiple dataset for training different models which is used to form the combined model. Generally, ensemble learning can be classified into two categories: sequential ensemble learning and parallel ensemble learning. Sequential ensemble learning utilizes the strong dependency of models, through giving a high weight of formal mislabeled sample to enhance the predict accuracy. Parallel ensemble learning generates models at the same time, through average the entire model's error can be reduced obviously. In a word, ensemble learning methods use multiple models to increase the performance of final output. Ensemble learning methods can be classified into three categories: bagging, boosting and stacking.

Bagging is a typical ensemble learning method, it has three features: (1)every subset is random sampled with replacement; (2)the final output is generated by voting strategy of all sub-models(classification) or the average of all sub-models(regression); (3) all sub-models are generated in parallel which are independent of each other. The representative bagging algorithm is random forest which using decision tree as bass model.

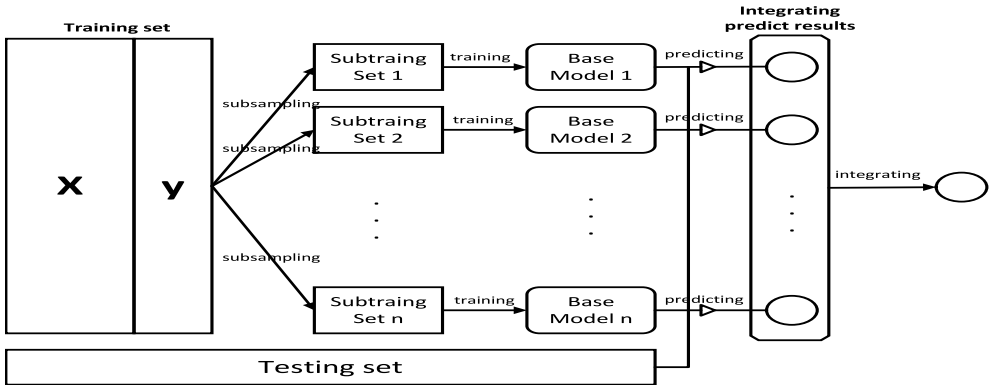


Figure 4: Bagging method

The implementation steps of random forest can be described as follows:

step 1: random sampling with replacement of the original dataset, getting n sub-training dataset. Each sub-dataset is used for training a single decision tree, thus there will be n different decision trees generated.

step 2: random selecting m features from feature set M (m < M) to construct a decision tree. The algorithm used for constructing decision tree is CART algorithm which using information entropy to select splitting node. The decision tree does not take any prune strategy.

step 3: using voting strategy or average strategy for the n generated outputs of decision trees to generate the final output of random forest.

The parameter m is the key point of the algorithm as it influence the depth of the tree and the classification accuracy of each decision tree. If m is too small, the classification accuracy of random forest will be lowered; however if m is too large, the model face the problem of over-fitting. It is

obviously that the number of trees n will influence the performance of random forest. Thus, during the training period, the task of random forest is selecting the two parameters m and n.

Boosting method concatenates multi-models to form the final model which means the output of previous base model will influence next model. The entire dataset will be used for training every base model. The misclassified data in previous model will be given a large weight for next model's training. This process will be repeated until the output of the model over predifined error rate or the iteration times is used up. The implementation of boosting model can be defined as follows:

step 1: initializing weight samples in training set, every sample is given same weight at the beginning.

step 2: training base models. If a simple is classified correctly, it will be given a lower weight, otherwise it will be given a larger weight. The new training set will be used for the new round training of base model. At the same time, based on the classification of accuracy, the base model will have a weigh either.

step 3: the final model is constructed by all base models with different weight. If the classification accuracy of a base model is low, the output of this model accounts for a small proportion of final output and vice versa.

If the base model of boosting method is decision tree, it will derive many famous machine learning algorithms such as gradient boosting decision tree (GBDT), XGBoost and LightGBM.

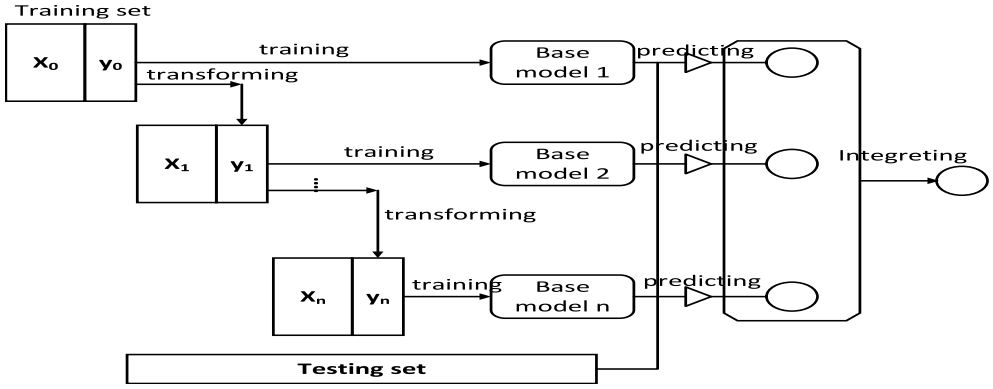


Figure 5: Boosting method

Stacking is using a model to combine other models. Different from bagging and boosting which integrating base models in a simple way, stacking uses outputs of base models as the input to train a classification model for the final task. The implementation of stacking can be described as follows:

step 1: random sampling with replacement of training set to construct n sub-training set.

step 2: using n sub-training set to train n models. The outputs of these models are the input for the classification model.

step 3: the final model is constructed by two level models. The first level includes n base models and the second level includes one classifier for the final task. Usually, the integration model is

logistic regression.

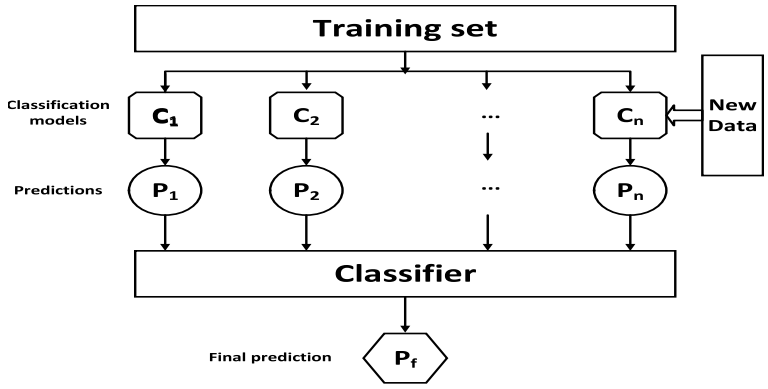


Figure 6: Stacking method

2.3 Deep Learning

Artificial intelligence(AI) is one of human's beautiful dream for long time. The concept was first proposed by Turing since 1950. However, the research did not get any progress for many decades due to the limitation of hardware for computing the complex network parameters. Since 2006, the research in machine learning break through the obstacles by cloud computing technology and the progress of algorithm. The algorithm is deep learning which is also called deep neural network. Deep learning can be understand as a subpart of machine learning which has good performance on the task that extracting features from raw data is difficult, thus this algorithm has another name called unsupervised feature learning. Before the generating of deep learning, the dominant algorithms in machine learning are support vector machine(SVM) and boosting. Feature extraction is the most important step for these algorithms and it influences the performance of the algorithm. Thus, machine learning can be thought as a subpart of representation learning, each problem can be described as using a set of features to build a model for the task of predicting. All in all, artificial intelligence is a broad topic which includes many many research fields, machine learning is just a subpart of it.

Machien learning has obtained good performance on structured data for long time. However, when the data is abstracted which is difficult to extract features such as image, sound or language, traditional machine learning algorithms perform not that good as human need to extract features first. However, feature extraction process by human is complex and inefficient, hence made the application of machine learning algorithms in these areas not prevalent. The emerging of deep learning changes this situation dramatically as it can extract features from raw data automatically by different layer of network. The shallow layer extracts basic features which can be further combined as structured features by higher layer network. For example, when an image is offered to a deep neural network, the shallow layer of the network will extract many basic features of the image

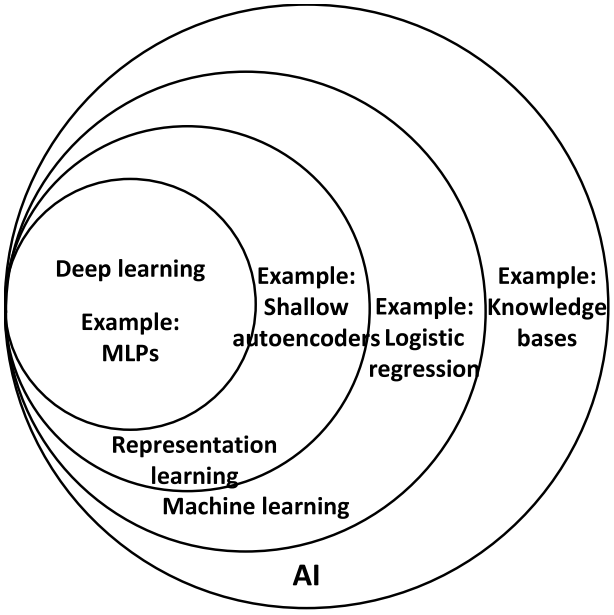


Figure 7: A diagram showing the relationship of deep learning, representation learning, machine learning and AI.[3]

such as small edges, these edges will form structured features like corners in the middle layer of the network, then these structure features will form skeleton maps of the image in deep layer. Finally, the network will make prediction based on these skeleton maps. Deep neural network extracts features in the hidden layer which simplified the complex operation of human, thus making it become the dominator of unstructured data area.

2.3.1 Fundamental of neural network

Neuron

Similar to our brain, neuron is the basic component of neural network. In neural network, a neural can accept one or multiple inputs and output processed results to the next neuron. If the neuron is located in the last layer of the network, its output will be the final result of the entire network.

Weights

When data as inputs to a neuron, it will be allocated a weight for each input. Inputs multiply weights is the input information of data to a neuron. The weights will be initiated at the beginning of training, then they will be adjusted with the iteration of training. The data with high weight is deemed as important data, but if the weight closes to zero means that data has no influence on the neuron.

Bias

The inputs to a neuron include two part, data information and bias. Bias is added to inputs

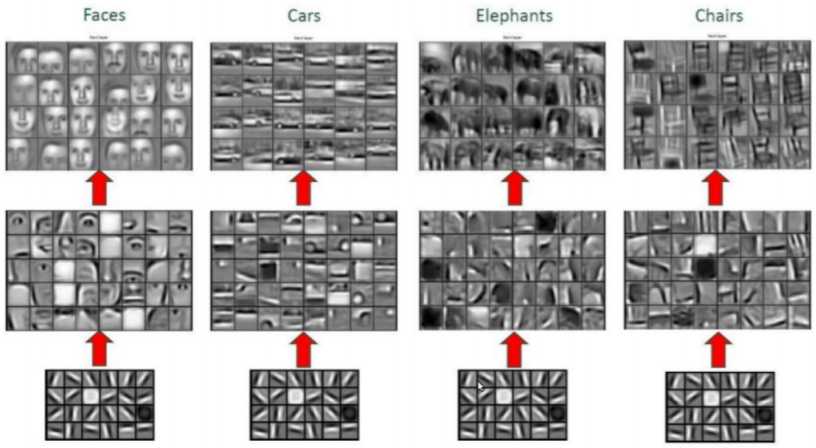


Figure 8: Hierarchical representation[3]

multiply weights. Bias can make the function fit data better and solve the problem caused by all inputs equal to zero. Figure 9 shows a neuron with 3 inputs, the output of this neuron is $x_1w_1 + x_2w_2x_3w_3 + b_1$, where x, w and b are all n dimensional vector.

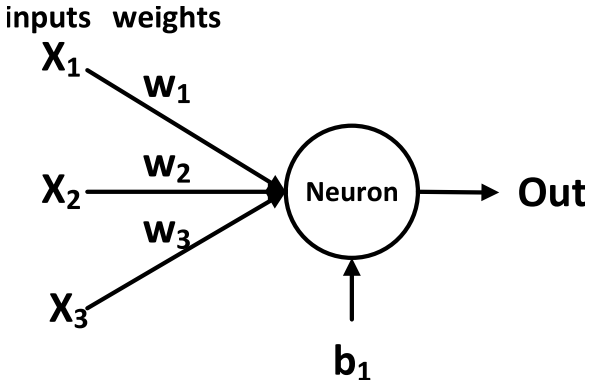


Figure 9: A neuron in neural network

Activation function

Activation function is applied on the output of neuron, the purposes of using activation function are: (a) achieving the nonlinear transformation of data to enhance the analyzing capability of model; (b) realizing data normalization by mapping it to a fixed range, this can limit data extension and prevent data overflow. The output after activation function is $f(xw + b)$. The most widely used activation functions are:

(1) sigmoid function:

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Sigmoid transformation generates a number in the range between 0 and 1. When the input is a very large negative number the output is close to 0, and when the input is a very large positive number the output is close to 1. Sometimes, we need to observe the output when the input changes, this function is a smooth curve which makes it perform better than step function. The disadvantage of this function is when the input is a very large negative or positive number, the gradient of this function is close to 0. This will cause a severe result during back propagation process call vanishing gradient.

(2) ReLU function:

$$\phi(x) = \max(0, x) \tag{2.2}$$

ReLU is one of the most widely used activation function in deep learning as it avoid the gradient vanishing problem in sigmoid function and realize the nonlinear transformation for neuron output. When the input to a ReLU function is negative, it will become 0. And when it is positive, the number will be output directly without any change.

Input, output and hidden layer

Input layer is the first layer of neural network, it accepts raw data as input of the entire network. Output layer is the last layer of neural network, the output of it is the final output of the neural network. Hidden layer is the layer between input layer and output layer, it receives signals from prior layer and passes the processed data to next layer.

Forward propagation

Forward propagation is the movement of data that being transferred from input layer to hidden layer, and output from output layer. This process is a one-way movement which outputs the processed data by the network.

Cost function

The purpose for training a model is making the output of the network as close to the real number as possible. Cost function is used for measuring the accuracy of the network, it will punish the network when mistake happens. To achieve the goal of increasing predicting accuracy and minimizing error, we need to minimize the output of loss function.

Gradient descent

Gradient descent is the optimization algorithm to minimize cost function which can find best parameter for a model. It is used to find the optimal solution for a function. Unfortunately, this method can only find local optimal solution unless the function is a convex function.

Learning rate

Learning rate decides the speed of cost function to approach optimal solution. The choice of learning rate is an art for cost function. If this value is too large, we may never find the optimal solution as it will be jumped over constantly. However, if it is too small, the time for reaching optimal solution may be too long.

Back propagation

The weights and bias will be initiated at the first stage of training a neural network. After first forward propagation, error of the trained model will be generated. This error and cost function will become the feedback to the trained network for updating parameters. This process is from output layer to input layer, thus being called back propagation.

Batches

Due to the large size of dataset, it is impossible to feed all data to a network at one time. The practical way to solve the problem is splitting dataset into many small blocks. Every time, a small block data, as a batch, is feed to the trained network. This process will repeat many times until all the data have passed the network.

Epoches

As said before, the dataset is split into many small batches. When a batch as the input to the trained network, it need to finish forward and back propagation. After all data finish forward and back propagation for once, an epoch is finished. The amount of epoch can be decided according to different requirements. If it is too small, the accuracy of the model may be lower than predefined threshold. However if it is too large, the training time may be too long.

Dropout

Due to the strong learning capability of neural network and the different distribution between training set and testing set, overfitting is a common phenomenon in deep learning algorithms. Dropout is a technique to prevent overfitting. During the process of training a network, some neurons in hidden layers will be dropped out randomly. Parameters with dropped neurons will not be updated, hence the trained model will not over fit on training set .

2.4 Feature engineering

There is a saying that data and feature decide the up bound of machine learning, good model and algorithm just try to approach this up bound. Feature engineering is the process of extracting features from raw data, the extracted features can be used to build a model which has good prediction performance. Good features can represent inner structure of data, thus most models can get good performance based on good feature set. According to practical experience, if the selected feature set good enough, the requirements for model selection and parameter adjustment will not be that strict. Compared with complex model, simple model with good feature set can get the same performance. Hence, most companies prefer to use simple models to solve problem as these models are easy to maintain and can be interpreted clearly. The experiment result is influenced by many factors such as model selection, data quality and feature selection. Even the equation to estimate the performance of a model can influence the experiment result, what we can do is finding a feature set to describe the inner structure of data.

With the development of technique, using models to extract features automatically becomes popular. Factorization machine and deep learning are prevalent models for feature selection. The intermediate results of these models can be used as the input to other model, for example, the output of hidden layer of a neural network can be used as input to logistic regression. According to

online testing results, the extracted features get good performance for predicting task.

To construct a feature set, we usually start from practical problem. The first step is finding related factors. Then, we need to conduct feasibility assessment on these features such as the difficulty for obtaining, degree of coverage for data, correctness etc. There are many things need to be taken into consideration, these complex operations are called feature engineering.

After extracting features from raw data, we cannot use these features as they still have many problems:

- The dimensions of different features are different, making the comparison of features impossible. Nondimensionalization is an effective way to solve this problem.
- For some quantitative features, effective information is included in divided interval which may result in information redundancy problem. For example, the student academic record is a numeric feature which can be transferred into binary feature representing failed and not failed.
- The qualitative features cannot be used as input to machine learning algorithm, hence they need to be encoded first before enter algorithms.
- Some features may contain missing values. Only after all missing values of a feature have been filled, can we use that feature to build a model.
- The way for different machine learning algorithms to utilize data are different. Extracted features need to be further refined to enhance the model performance. For example, if we plan to train a simple logistic regression model, we need to discretize continuous features first. Then, using one-hot encoding to encode these features. These feature processing steps improve the capability of a model to deal with nonlinear problems.

Feature engineering includes many sub-problems, these problems can be summed up as three stages of feature extraction: (1) feature cleaning, data cleaning; (2) feature processing, data processing; (3) feature selection. We will introduce these stages in detail in the sub-sections.

2.4.1 Feature cleaning, data cleaning

Usually, business data is imperfect which contains many problems such as reported exception, cheating behavior. In order to learn the pattern behind data, the first step is imputing data and removing dirty data. This includes two aspect:

1. filtering dirty data according to business requirements such as cheating data, spam etc.;
2. using outlier detection algorithm to find anomaly data, common anomaly detection algorithms are:
    - Measurement based on deviation such as k nearest neighbors, cluster.
    - Anomaly detection algorithm based on statistics like box plot.
    - Anomaly detection algorithm based on distance. When the distance between a point with other points is larger than a threshold, this point is deemed as outlier.
    - Anomaly detection algorithm based on density. When the density near a point is different from others, this point may be thought as outlier.

2.4.2 Feature processing, data processing

Generally speaking, features can be classified as several types: continuous feature, discrete feature, time, text and combined feature. Different feature has different strategy to deal with, below is a brief introduction.

1. Continuous feature. As the range of a real number can be very large, it is hard to build a model with features in different range. Besides, sometimes we only care the meaning of a feature in a fixed range rather than the detailed value. There are two common ways to deal with continuous value:

   - Normalization. Normalization can solve many problems such as increasing the speed for finding optimized solution, increasing the precision of model and making its possible to compare different features. Normalization may sacrifice some information of feature, but compared with the benefits this small loss is acceptable.
     Normalization can be classified as three types:
     Linear normalization is suitable for centralized data, however, if the maximum and minimum value are unstable, other value will also be changed.

     $$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2.3}$$

     Standardization is assuming that random data follow normal distribution.

     $$x^* = \frac{x - u}{\sigma} \tag{2.4}$$

     Nonlinear normalization is using a math function to map data before normalizing. The selection of mapping function is based on practical requirements, common mapping functions include log function, exponential function and tangent function. These normalization method can fit better on business requirements.
   - Discretization. Discretization is a process of transforming continuous feature to discrete feature, this process is also called binning. Binning is pretty useful for linear model as it shifts the linear relationship between a feature with the target to nonlinear relationship. The way to realize this is assigning a weight to every discretized feature component. According to experiment results, discretization largely improves the fitting capability of a model. The common discretization methods includes constant frequency discretization, constant distance discretization and tree model based discretization.

2. Discrete feature. For category feature such as grade which is separated into 5 level(A-F), the common way to deal with this type data is transforming every attribute into binary attribute i.e. taking a value from 0,1. Thus, the number of increased feature equals to the number of category. For every instance of data, only one position is marked by 1 of the encode feature. This feature encode process is called one-hot encoding. For example, if a student get A in an exam, the feature value of grade for this student is 10000.

3. Time. Time can be thought as a feature type which is very common in different data set. It can be continuous feature like the time to stay at a web page, or discrete feature like weekdays.
4. Text. Text is a special feature which is hard to deal with compared with other features. The common ways to deal with text feature are: bag of words model and word embedding. With the popular of deep learning, there is a open tool word2vec which can transfer text into word vector. The generated word vector is suitable for feeding in deep neural network.

Besides, we can construct feature by combining different feature together. Sometimes, the constructed feature could be a windfall.

As said before, business data is pretty dirty which may contain many missing values. Commonly, we will fill the missing value by average value or a default value. But, if the missing value of a feature take a large proportion, this feature may be dropped directly. Feature processing is a complex process which needs a lot of patience, but it is important as the quality of feature defines the up bound of model performance.

2.4.3 Feature selection

Usually, the processed features cannot be used for model training directly as the performance of the trained model is not optimal. The reasons for feature selection are: (1) the feature dimension is too high which may contains redundant features for training a model; (2) some features are highly correlated, this consumes too much computational resource; (3) some features may introduce noise which has negative influence the performance of trained model. Hence, the purpose of feature selection is finding the best feature combination for training a model. There are two aspects need to be taken into consideration when conducting model selection:

- The divergence of feature. If the variance of a feature is close to 0, this feature cannot distinguish samples from each other.
- The correlation between a feature and the target. The more a feature correlates with the target, the more likely it is selected for taining a model.

There are some interaction between features, some feature may include other feature, some feature need to cooperate with other feature to use, some feature have negative correlation with other feature. Hence, feature selection play an important role on model training. Generally, model selection method can be classified as three types: filter, wrapper and embedded. Feature selection can increase the generalization capability of a model, reduce the amount of feature and reduce the risk of overfitting.

1. **Filter** is used for measuring the correlation between single feature with target. But it does not take the correlation between features into account, some important feature which need to combine together with other feature to use may be wrongly filtered out. There are many metrics can be used for filtering out data such as chi-square test, mutual information coefficient and pearson correlation coefficient. The other idea for filtering is using the statistical characteristic to remove features which have no distinctiveness. This method is only suitable for discrete feature and continuous feature need to be discretized before applying this method.

There are some machine learning models can be used for feature selection. After applying the model on entire feature set, the importance of feature is ranked by the model automatically. For example, random forest uses information gain to select features for tree construction. Features with large information gain will be placed at higher layer of a tree, thus a built forest has finished the ranking of all features according to information gain.

2. **Wrapper** is ideal feature selection method in theory, but it does not have many practical value in industry. Fundamentally, wrapper is an iteration process. Recursive feature elimination(RFE) method use a base model for training, features with small weights will be removed after each round. This process will repeat many times until the size of feature set satisfied requirement. Another classical method is RFECV which tries all possible subsets of feature set. It calculates validation error of based model for every subset, the subset which make base model getting smallest validation error will be selected as final feature set.
3. **Embedded** method selects multiple features at a time. For example, using a feature set to build a logistic regression model with L1 regularized penalty term, most feature weights will be 0. Features with nonzero weight will be selected for model training stage. Besides, gradient boosting decision tree (GBDT) is widely used for feature selection.

All in all, the purpose of feature processing and selection is choosing best features for model training. The questions is how to estimate a feature? Usually, we need to think from two aspects: (a) The quality of a feature itself. A good feature should not contain too much missing value or wrong data. If a feature cannot discriminate samples from each other, this feature should be filtered out from feature set. (2) The correlation between feature and target, and the correlation between features. Besides, when it comes with business data, there are many classical strategies for business requirements. For example, weight of evidence (WOE) and IV value are widely used for measuring the importance of single value. Variance inflation factor (VIF) is used for measuring multicollinearity problem between features. Population stability index(PSI) is used for measuring the stability of model.

Sometimes, the selected features cannot be used for training a model due to the large dimension of feature set consumes too much resources. If the computing resource is limited or the training time is too long, the dimension of feature set must be reduced. Principal component analysis (PCA) and Latent Dirichlet allocation (LDA) are commonly dimension reduction methods. Both of them focus on proportion the feature set from a high dimensional space into a low dimensional space. PCA is a unsupervised dimension reduction method which tries to increase the divergence of features. But LDA is a supervised dimension reduction method which tries to increase classification accuracy of samples.

2.5 Imbalanced learning

Financial fraud detection is a task which tries to find black samples from imbalanced dataset where the amount of white samples far more than black ones. Imbalanced learning is a solution for this problem which mainly includes two methods: random sampling and loss-sensitive learning. The

difficulty of imbalanced learning is data complexity. There are many reasons influence the performance of classifier[9]: 1.The distribution of data belong to different class can be overlapped, trained classifier is hard to separate the overlapped dataset. 2. In industry scenery, it is inevitable to introduce noise data, these noise data will influence the result of classification. 3. For the minority class, samples in this class cluster several sub groups. Thus, the binary classification problem is indeed a multiclassification problem. Due to above reasons, imbalanced learning is not an universal solution for data distribution imbalance problem.

2.5.1 Random sampling based methods

Random sampling focus on solving dataset imbalance problem. It tries to balance the amount of samples in different class. Ramdom sampling methods include oversampling, downsampling and synthetic sampling.

Oversampling takes samples from minority class, and puts a copy of every sample back into the original class. This process will repeat many times until the distribution inclination problem of original dataset is solved. Downsampling takes out samples from majority class. The purpose of downsampling is also try to balance the amount of samples in different class. The difference between oversampling and downsampling is that oversampling is enlarge the size of minority class while downsampling is remove samples from majority class. After oversampling or downsampling, the size of different class reach balance. However, this balance just looks like balance, different sampling strategies will influence the performance of model[10]. When using downsampling method, removing large amount of samples from majority class may result in the loss of important information for classifier training. Compared with downsampling, the problem caused by oversampling is more serious as the copied samples will increase the possibility of overfitting for classifier training. In order to overcome the inherent drawbacks of undersampling and oversampling, a lot of methods have proposed such as EasyEnsemble and BalanceCascade[11].

2.5.2 Cost-sensitive learning

Different from random sampling method, cost-sensitive learning accept the premise that data distribution is imbalanced. It focus on the cost caused by samples which are miss-classified. In real business activity, the cost caused by a fraud activity is much higher than the cost caused by preventing a normal activity. If the fraud activity is succeed, the loss to an organization is irreparable. However, if a normal activity is prevented, it can by solved by redo it and apologize to the customer. Cost-sensitive learning defines a cost matrix to represent the cost of miss-classification. There is a research shows that cost-sensitive learning has strong correlation with imbalance learning, the theory of former method can be migrated to later one and this has got a lot of positive outcome[12].

3 Related Work

Financial fraud is not a new topic, it is economic crime which can cause serious loss to a financial organization. According to the statistics given by Association of Certified Fraud Examiners, the average loss caused by fraud of an organization is 5% of its annual revenues[13]. Hence, financial organizations have taken a serious antifraud measures to confront fraudsters. In early stage, frauds are primarily detected by expert auditors which need to get long period training by corporation. For example, Weisberg and Derrig hired trained claim adjusters to identify the responsibility of accident drivers and suspicious claims[14]. However, with the astonishing development of technology and the largely improved life quality, there are enormous amount data generated everyday. Hence, traditional manual audit method for fraud detection is insufficient. Researchers and industry corporations have tried many automatic methods for fraud detection, actually, some of them have got great success and have been applied in industry use as the first prevention for financial fraud detection. At beginning, statistic methods have been used for analyzing this problem such as Markov model. With the prevalent of machine learning algorithms, there are many attempts for using machine learning algorithms to solve fraud detection problems such as clustering algorithms, support vector machines (SVM). Among them, logistic regression(LR) and gradient boosted trees(GBT) are the most popular methods for financial fraud detection which is usually used as baseline in industry use. Recently, deep learning is found as a powerful tool in many fields such image recognition[15] and natural language processing[16]. These fields have a common characteristic that is extracting features from these data is difficult. The deep layer structure of neural network can extract complex features automatically from the raw data which make deep learning algorithms outperform other machine learning algorithms in these fields. Different from unstructured data, financial data is typical tabular data which means that each dimension of a sample can be deemed as a feature. The difficulty of feature engineering is not feature extraction but feature selection. Existing machine learning algorithms have got good performance in this field, but the charm of deep learning still induce many researchers to apply deep learning algorithms in financial fraud detection.

3.1 Financial fraud detection

Adrian Bănărescu[17] conducted a comprehensive analysis on using data analysis methods for detecting and preventing fraud. It compared data analysis softwares such as Microsoft Excel, Microsoft SQL Server. Besides, it classified data analysis methods into two categories: operational analysis and strategic analysis. The operational analysis, such as risk analysis, results analysis etc., is suitable for detecting frauds happened recently. While the strategic analysis, such as case analysis, SWOT analysis etc., is suitable for macro-level analysis. The author stressed that for fraud detection, proactive detection is much more useful than negative detection. As the loss caused by fraud is irreparable,

prevention is much more important than make up the loss. Besides, if a antifraud method can only detect frauds happened before, this method is too week as treacherous fraudsters will try their best to find new loopholes of the system.

Nowadays, digitalization is not only benefit human's daily life, but also enriches the technique for fraudster to launch an attack to a system. Financial institutions continuously improve feature engineering and techniques to confront skimming fraudsters. Rushin et al.[18] thought that a business organization need to optimize fraud detection under business and customer constrains. If the fraud detection rule is over strict, many normal business behavior may be intercepted which will decrease the user experience. However, if the threshold is defined too large, fraud detection system may miss some suspicious behaviors which will serious loss to the financial organization. They compared three supervised learning models: logistic regression(LR), gradient boosted trees(GBTs) and deep learning. Actually, logistic regression and gradient boosted trees(GBTs) have been widely used in business organization and got good performance. However, based on their experiments, deep learning outperform these models which show a potential choice for financial fraud detection.

Schreyer et al.[19] applied deep learning techniques for detecting anomalous journal entires in large scale accounting data. They proposed a deep network structure which is similar to autoencoder name AE network. Inorder to fully explore the capability of this network, they proposed a series networks with different number of layers and the best performance network is achieved by the deepest network. Besides, they compared the deep learning models with other non-parametric anomaly detection techniques(Principal Component Analysis, One Class Support Vector Machine, Local-Outlier Factor and Density-Based Spatial Clustering of Applications with Noise), and got the conclusion that deep learning models outperform other models in terms of prediction precision. In order to verify the effectiveness of proposed model, they applied the model on two different dataset to test its performance. The advantage of this model is that it can detect unknown fraud behavior which is not derived from known fraud scenarios. Until now, the boundary of deep learning for fraud detection has been further broadened.

Compared with numerical and categorical data, text data contains rich information and is hard to be forged by experienced deceiver. However, it is hard to utilize hidden information behind it by traditional machine learning classifier such as random forest and support vector machine (SVM). Wang and Xu[20] proposed a LDA- and deep learning-based automobile insurance fraud detection model. This model utilizes the text information in automobile documents which is extracted by Latent Dirichlet Allocation(LDA) algorithm. A deep neural network(DNN) is utilized to solve the fraud detection problem. This model is a supervised classification model, the input sample to this model consists of 10 categorical attributes, 5 numeric attributes and 1 text attribute and the output is an indicator which represents the class label. Via comparative experiment, the authors verified that deep neural network with text features largely outperform the same network which is only inputted categorical and numeric data. This thesis exhibits the power of deep learning networks to deal with complex data. The multielement input to deep learning network makes deep learning algorithm outperform other machine learning algorithms.

Business behaviors are various, different behavior has its inner characteristic. Hence, different

machine learning algorithm has its own area where it can use successful results to maximum advantage. For example, online E-Commerce transaction is a series click events which is suitable for time sensitive algorithms. Wang et al.[21] proposed a novel approach to capture detailed user behavior in purchasing sessions for fraud detection in e-commerce websites. Using recurrent neural network(RNN) to capture the sequence of clicks, revealing the browsing behaviors on the time-domain. The RNN model is super suitable for online business problems as the customer behavior is a series clicks which has has hidden order in it and RNN model can make full use of this features for prediction. Besides, as the click behavior is hard to represent, they introduced the Item2Vec[22] idea to encode click events which can reduce the sparsity of input vector. This research also offers a view about the deep learning structure that is wider networks usually provide better memorization ability, while deeper networks are good at generalization.

Chouiekh et al.[23] applied deep learning techniques in mobile communications and the dataset comes from customer details records (CDR) of a real mobile communication carrier. A real time fraud detection system based on deep convolution neural networks (DCNN) was proposed. The DCNN moder is compared with traditional machine learning algorithms (support vector machine, random forest and gradient boosting classifier), the performance of DCNN model is better than other algorithms in terms of accuracy. Due to deep learning networks use GPU as computing resource, the training duration of DCNN is remarkably faster than other classifiers. This experiment result underpins that the proposed DCNN model is suitable for real time fraud detection which needs fast reaction when unforeseen circumstances happened.

Although, there are many researches on applying deep learning models for fraud detection, seldom people analyze the influence of deep learning topology on prediction accuracy. Roy et al.[24] conducted a comprehensive research on evaluating different deep learning topologies with regard to their efficacy in detecting credit card fraud. It analyze various parameters that are used to construct the model to find the optimal combination of parameters to detect fraudulent activity. Feature engineering and dataset imbalance problem has been solved in advance. The deep learning topologies used for analysis are: artificial neural networks(ANNs), recurrent neural networks(RNNs), long short-term memory(LSTMs) and gated recurrent units(GRUs). Their research got a lot of meaningful conclusion for applying deep learning algorithms on financial fraud detection:

1. The size of the network is the largest driver of model performance. Large network tend to perform better than smaller networks.
2. Hyperparameters, such as momentum, have less impact on the model performance except learning rate. For GRU and ANN the best performance learning rate is 0.05 while for LSTM and RNN the best performance learning rate is 0.5.
3. Loss functions are comparable. The binary cross entropy and consine proximity loss function lead to a much wider range of model accuracy scores.
4. LSTM and GRU significantly outperform the baseline model ANN which indicates that an account's transaction order contains useful information for differentiating fraud and non-fraudulent transactions.

Deep learning algorithms learn intermediate concepts between raw input and target. It use multiple non-linear processing units for feature extraction and transformation. The author also posted that the number of neurons in each layer may also reveal additional insight into the effect of network size on model performance.

3.2 Dataset imbalance

Dataset imbalance problem is an inevitable problem in fraud detection. There are paramount researches on this field and most of them have got good performance such as SMOTE sampling method. But using deep learning models to solve dataset imbalance problem is a relatively new direction.

Fiore et al.[25] trained a generative adversarial network(GAN) to output mimicked minority class examples which will be added into training data lately to form an augmented dataset. Experiments show that classifier trained on the augmented set outperforms the same classifier trained on the original data, especially as far the sensitivity is concerned, resulting in an effective fraud detection mechanism. It compared the classical oversampling method SMOTE with the proposed model, based on experiment result, the proposed model performed better. However, this model can only be used for detecting frauds which has shown in training set. For those which never been detected by the system, it is hard for the proposed GAN model to detect them.

With the fast development of online business, online fraud detection technology plays an important role in identifying fraud cases, recovering losses and avoiding risks for customers and online platforms. Fraud detection in this area also need to pay attention to big data problem as online business will generate enormous amount data. Besides, the dataset imbalance problem will be more serious. Gao, Yang[11] design and implemented a online fraud detection system using big data processing technology, including Spark[26] and Spark Streaming[27]. It improved two imbalance learning methods for fraud detection: (1) An incremental clustering-based dataset self-balancing construction algorithm is proposed to measure the similarity of intra-class samples, choosing representative samples. (2) A distributed loss-sensitive Lasso algorithm is proposed, which can efficiently learn the model in the context of big data and effectively improve the loss rate of assets. In order to get good performance, the author also made some effort on feature engineering, he introduced derived predictors based on expertise knowledge such as frequency of transaction per month. It is obvious that no matter in which application scenario, dataset imbalance is a tricky problem need to be solved.

3.3 Feature engineering

Another inevitable task in financial fraud detection is feature engineering as financial data usually have many attributes which represent different indicators. Not every feature will contribute to the classification model. Sometimes, we need to derive features by manual. The more careful in feature engineering stage, the better performance can be obtained for a classifier.

Rushine et al.[18] in their thesis proposed an autoencoder to extract features automatically. It is an unsupervised feature engineering method which can reduce the dimensionality of data and

slightly boost predictive power. In order to verify that whether the proposed feature engineering model can replace traditional manual work. The author compared the classsifier's performance with feature set extracted by domain expertise and autoencoder. The experiment result show that the latter performed worse than the former. Thus, it is impossible to use automatic feature engineering method to substitute human work at present stage. But it is deserved to conduct more analysis on feature engineering method, such as principal component analysis and random forest, which can help to reduce the heavy burden of human and deep learning algorithms is a good research direction.

Wang et al.[20] pointed out that fraud indicators play a critical role in insurance fraud detection. Appropriate indicators definitively make it possible for detection methods and algorithms to maximize the effectiveness of detection. As we said before, features(indicators) extracted by domain expertise can largely improve the performance of classifier. There are some universally accepted indicators that can help boost the performance of classifiers. Some scholars have sorted these indicators into several groups, for example the accident information, the insured driver information and the automobile information. This paper also conduct an in-depth analysis of numerical and categorical fraud indicators such as time, location. However, structure data is easy to counterfeit by experienced deceivers, therefore, text data should be taken into consideration for building a robust feature set.

Kazemi et al.[28] propose a deep autoencoder to extract best features from the information of the credit card transactions and then append a softmax network to determine the class labels. The output of encoder part of autoencoder network is deemed as extracted features by the proposed model. The best performance network is obtained by a structure, called sparse autoencoder, which the number of neurons in hidden layer is much more than input and output layer. The author also compared sparse autoencoder with the converse structure which neurons in deeper layer is less than other layers, the performance of sparse autoencoder is better than the latter one. This thesis confirm that the wider of deep learning network also has great impact on the performance in practical scenario.

4 Choice of methods

This part will illustrate methods chosen for answering research questions proposed in this thesis. Figure 10 gives an overview of the methodology used for fraud detection. The structure of this chapter follows the data processing procedure: dataset selection, feature engineering, countermeasure for dataset imbalance and selection of classifiers.

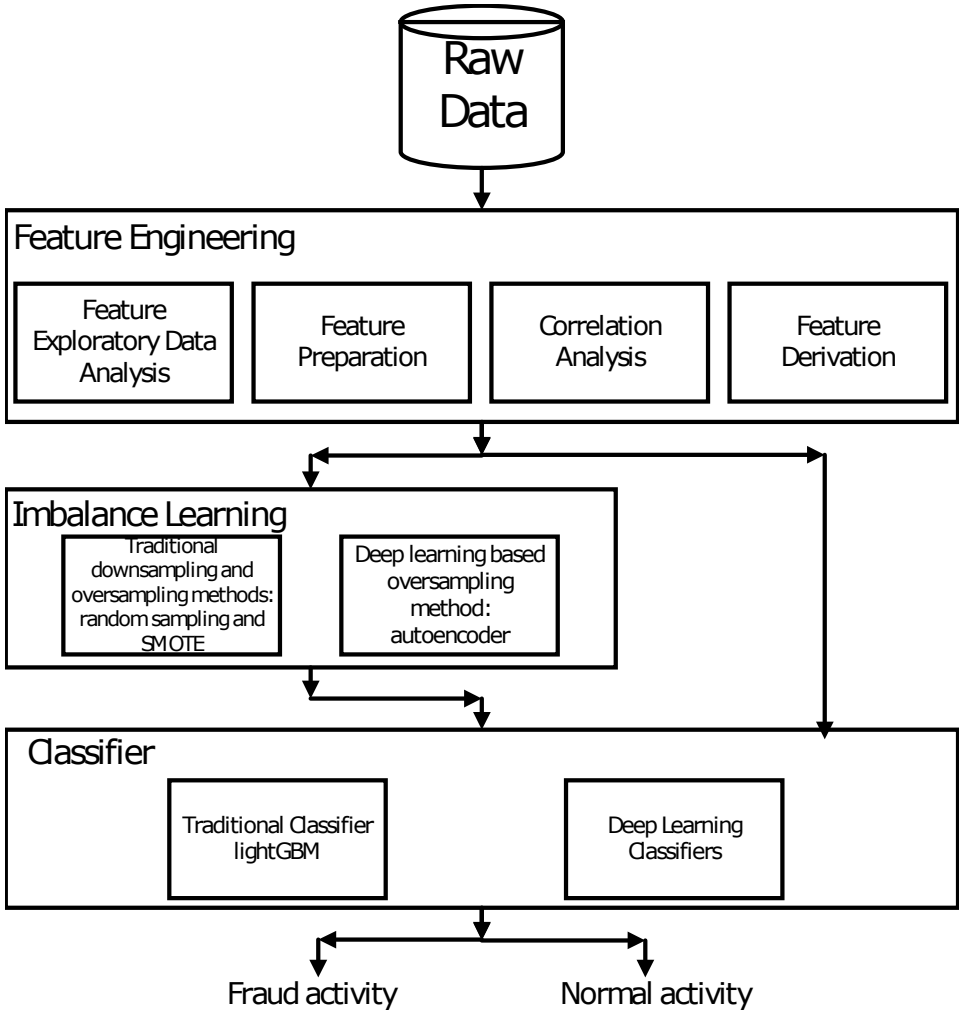


Figure 10: Methodology

Data becomes a more and more expensive resource in recent years. Many corporation would like to spend a lot of money on obtaining users' data. Due to the sensitivity of financial data, almost no organization will share user data with other organization. Thus, It is extremely difficult to obtain financial data to conduct experiments. Luckily, there are some public datasets offered by some competition can be utilized. As deep learning needs large data to train network parameters, this thesis chooses a relatively large dataset comes from IEEE-CIS Fraud Detection competition. Attributes in this dataset related to privacy are encrypted. After obtaining dataset, the first step for financial fraud detection feature engineering. There are many techniques and methods can be used in this stage, this thesis only explains methods used in our experiments, including feature exploratory analysis(EDA), feature preparation, correlation analysis and feature derivation. Sometimes, samples after feature engineering can be fed into classifier directly. However, due to the specialty of financial data, dataset imbalance is an inevitable problem. This paper proposed a deep learning based oversampling method to solve this problem and compared it with another famous oversampling method SMOTE. To conduct a comprehensive analysis on using deep learning models for fraud detection, we tested several different deep learning models and compared their prediction results with the base model lightGBM[29]. Finally, to answer the question about using deep learning models to simplify feature engineering process, we compared the performance of deep learning classifier with different input feature set: the one is feature set with derived features(manipulate by use in feature engineering stage) and the other is feature set without derived features. In the following subsections, we will explain these methods in detail.

4.1 Dataset

Financial data has large commercial value and is highly related to personal privacy. Hence, it is extremely difficult to obtain financial data from business organizations. Luckily, there are some public competitions which are aimed to improve financial fraud detection strategies. IEEE-CIS Fraud Detection is one of the most classical financial fraud competition. It is held by IEEE Computational Intelligence Society (IEEE-CIS) and Vesta Corporation. The data offered by this competition comes from real-world e-commerce transactions. The merit of this dataset is that every sample has rich feature information which is similar to practical situation in industry. The goal of this competition is fully tapping the potential power of machine learning models for fraud detection. A success model will improve the efficacy of fraudulent transaction alerts and help tens of thousands of organizations reduce their fraud loss.

IEEE-CIS focus on AI and machine learning areas, including deep neural networks, fuzzy systems, evolutionary computation and swarm intelligence. They take advancing nature-inspired computational paradigms in science and engineering as their duty, try to be the international leader in innovative interdisciplinary research, technology transfer, real-world applications, and education in computational intelligence.

Vesta Corporation is the data provider of this competition. It is a world's leading payment service company which can be seen in many different credit card. It is the pioneer in guaranteed e-commerce payment solutions. Dating back to 1995, Vesta tried to apply fully guaranteed card-

not-resent(CNP) payment transactions for telecommunications industry. Since then, it expanded data science and machine learning technology in global business and solidified its hegemony in ecommerce payments. According to official statement, Vesta guarantees more than $18B in transactions annually.

To make sure that our porposed methods are powerful and reusable, we applied them on another dataset which has been widely used in many academic researches. It is a credit card transaction dataset offered by Worldline and Machine Learning Group of ULB[30–37]. This dataset is a relatively small dataset which is severly imbalanced, it contains only numerical features which can be utilized for distinguishing normal and fraud transactions.

4.2 Feature engineering methods

Feature engineering is an important step in data mining. There are many techniques can be used in this area[38] and some of them have been sorted as common procedures for feature engineering. This part will introduce feature engineering methods used in this thesis and the results of them will be shown in next chapter.

Exploratory data analysis(EDA) is usually used as first step for financial data exploration. By using data visualization, counting statistical values to find abnormal behaviors in data. This operation will help us to be familiar with the data we obtained. Main methods used for data exploration are: (1) data distribution visualization for training and testing dataset; (2) for binary classification problem, it is possible to visualize the distribution of different classes. Boxplot, histogram and statistical features are common tools for exploratory data analysis. After the processing of EDA, we can have a general image of all features. For e-commerce transaction data, there should be a timestamp feature which represents the transaction time. This feature offset seriously which influences the performance of other features. Hence, it will be removed from feature set for model training. But, this feature is useful for exploring the potential characteristic of other data. It usually be used as X-axis to make plot with other features. This will find features which have strong correlationship with time. There are many tricks can be used with this feature such as using other time related features to minus this feature to remove the influence of time correlationship. Besides, we will find that some features have good performance on differentiating normal and fraud behavior and some features have obvious difference on training and testing dataset. Furthermore, the statistical values can tell the missing value percentage, number of categories for categorical values etc.. All in all, exploratory data analysis will help us have a better understanding of data which will benefit for the following feature engineering procedures and model building.

Industry data is dirty, thus data imputation is an inevitable step for feature engineering. Generally, there are two types data need to be processed: missing value and outlier. If a feature has

large proportion missing value, we will consider drop this feature directly. Otherwise, replacing this value by another value such as mean/median/mode of that variable is the common method for these data. As for outliers, the first task is detecting outliers in data. Common methods for outlier detections are: IQR method, mean and standard deviation method. After detecting outliers, outlier imputation methods can be used such as replacing outliers by maximum and minimum of a distribution or discarding outliers directly.

For this dataset, there is an extremely important data analysis need to be conducted which is correlation analysis. Due to the number of features of IEEE-CIS dataset is very large, it is time and resource consuming to use all features to build a model. Besides, some bad features may decrease predict accuracy of the trained model. The idea of correlation analysis in this thesis is described as follows:

- Try to keep original features such as features started by character C, D and M.
- Use all data to calculate correlationship of features. Due to many features have missing values, those which have same number of missing value in same place should be put into same group.
- Set reasonable threshold for feature selection. Usually, the threshold is defined by classifier's performance. A series thresholds are proposed, and features selected by these threshold will be used for model training. A model with best predict accuracy decides the threshold of feature selection.

Tabular data can be classified into categorical and numerical data. In order to optimize the performance of the system, we take some strategies to reduce the memory occupation. All categorical features are encoded into numbers which are saved as integers in memory. Until now, almost half of the features have been removed but the model still has the problem of overfitting. From feature engineering perspective, the essence of overfitting is the difference of data distribution in different dataset. If offset of data in different dataset is over obvious, simply adjusting parameters of a model is useless. A classical method called confrontational verification is proposed for this problem. All data in training set are labeled as the same class i.e. 1, and data in testing set are labeled as another class i.e. 0. After training a model on the merged dataset, we can use prediction accuracy and feature importance analysis to find features which cause skewing of dataset distribution. The detected features can be dropped if they do not have much influence on prediction accuracy. However, some of them contribute a lot for detecting abnormal behaviors, deleting these features will cause obvious decrease of system performance. We choose some moderate methods to reduce feature offset: categorical features are transformed into continuous features and numerical features are shifted by logarithm function.

As the dataset contains more than four hundred features, the entire feature engineering process is complex and time consuming. But the result of feature engineering is desired and the performance of proposed model is improved a lot. We can draw the conclusion that feature engineering is a complicated but inevitable process for business intelligence area.

4.3 Strategies for dataset imbalance

In financial fraud scenario, dataset is seriously imbalanced. The number of normal customers is far more than fraudsters, hence the model may not capture enough features of fraudsters' behavior due to insufficient training data. The direct idea to solve this problem is generating synthetic fraud data, there are many oversampling methods and the most famous one is SMOTE. To balance the data amount of two dataset, random sampling the same number of data amount of minority class from majority class is another way to solve dataset imbalance problem. But this method may cause serious information loss due to too many useful data are dropped. In many situation, undersampling and oversampling methods are used together to balance the data amount in different classes. We proposed an autoencoder to generate samples in minority class. To test the efficiency of the proposed oversampling method, we compared it with SMOTE and random undersampling method. A baseline classifier is trained on these three dataset, the area under curve(AUC) value is used as the metric of system performance.

Ramdom undersampling method randomly selects the same amount of samples in minority class from majority class and removes other samples. This operation can reduce the dataset distribution imbalance problem, but may cause another serious problem which is information loss due to too many samples in majority class have dropped.

SMOTE is the most representative synthetic sampling method, and this method has obtained ideal results in many fields. Different from simply random oversampling method, SMOTE utilizes the similarity of minority samples' feature space to synthesize minority samples. The minority class $S_{\min} \in S$, K is the hyperparameter of K nearest neighbors. For every sample belong to the minority class $x_i \in S_{\min}$, firstly find its K nearest neighbors. Then, applying space transformation on one of its K neighbors which is selected randomly. The synthetic sample is composed of original sample with transformed neighbor.

$$x_{new} = x_i + (\hat{x}_l - x_i) * \delta \tag{4.1}$$

$\hat{x}_l$ is one of the K nearest neighbors which is selected randomly, and it belongs to minority class $\hat{x}_l \in S_{\min}$. δ is a random number in the range 0 to 1, $\delta \in [0, 1]$. Hence, the synthetic sample locates in between x_i and its neighbor $\hat{x}_l$.

Figure 11 illustrates the flow of SMOTE algorithm. Red crosses are samples belong to minority class and the hyperparameter of K nearest neighbors algorithm is four. After applying SMOTE algorithm, a synthetic sample is generated in between $\hat{x}_l$ and x_i which is represented by a black cross. These synthetic samples will release the problem caused by randomly oversampling and improve the performance of classifier. However, the generalization ability and anti noise ability of this method is imperfect[39], there are many problems need to be solved.

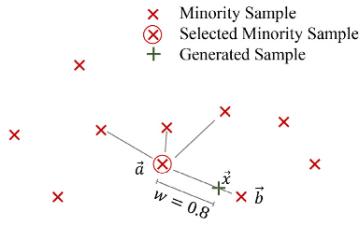


Figure 11: SMOTE linearly interpolates a randomly selected minority sample and one of its k=4 nearest neighbors[4]

Autoencoder(AE) is an unsupervised learning technique which we leverage neural network for the task of representation learning[40]. The typical application area of this technique is data dimension compression and feature expression. An autoencoder is composed of an encoder and a decoder, figure 12 shows the structure of an autoencoder:

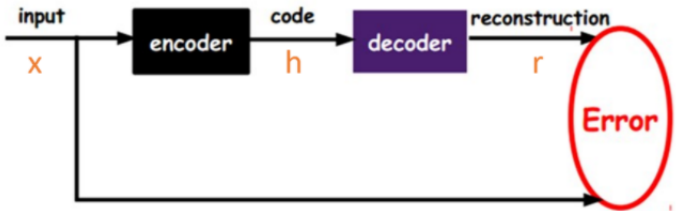


Figure 12: The theory of autoencoder

$h = f(x)$ represents the encoder of an autoencoder, $r = g(h) = g(f(x))$ denotes the decoder of an autoencoder, the target of an autoencoder is optimizing the loss function $L(x, g(f(x)))$.

In this paper, we tried two types autoencoder for synthesize samples in minority class, one is basic Autoencoder and the other is Sparce Autoencoder. The task of synthesizing samples can be understand as a supervised learning problem which the output $\hat{x}$ is a reconstruction of the original input x. This network can be trained by minimizing the reconstruction error $L(\hat{x}, x)$, which measures the difference between the original input and the generated sample. The bottleneck is a key attribute of the network design, it can be very simple structure which just memorize the input value by passing these values along through the network. Figure 12 is the structure of this type autoencoder, and we called it as basic Autoencoder.

We can see that when the number of neurons in the hidden layer is large than input and output layers, simply copying the input to output has a high possibility to happen. The bottleneck need to constrain the amount of information that can traverse the full network, forcing a learned compression of the input data.

An ideal autoencoder need to be sensitive to the input to accurately build a reconstruction, and be insensitive to the input so that the model will not just memorize or overfit the training data. In

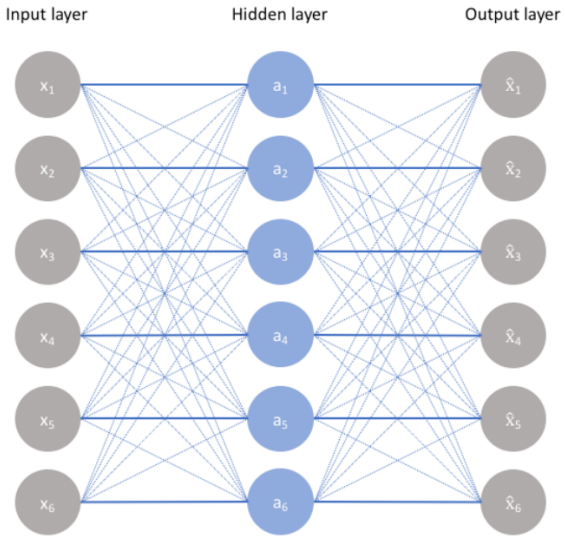


Figure 13: The structure of a basic Autoencoder

most case, this is realized by constructing a loss function where one term encourages the model to be sensitive to the input (reconstruction loss function $L(\hat{x}, x)$) and a second term discourages memorization/overfiting (an added regularizer). Usually, a scaling parameter is added in front of the regularization term so that we can adjust the trade-off between the two objectives.

$$L(\hat{x}, x) + \delta * regularizer \tag{4.2}$$

Sparse Autoencoder looks similar to basic autoencoder which does not reduce the number of nodes in hidden layers. The difference is that for any inputs, only a small part of neurons will be activated for training the network. This is a different approach for regularization as only *activations*(neurons which have been activated by inputs) will be regularized.

Figure 14 illustrates the structure of a sparse autoencoder where the opacity of a node represents the level of activation. It should be stressed that the activation level of each node in the network is decided by input data, different inputs will stimulate different nodes of the network.

Sparse autoencoder allows each node of hidden layer to be sensitive to specific attribute of the input data. Hence the activate regions of the network are forced to be a small part of neurons which are decided by input data. The memorization capability of the network is largely limited by *activations*, this makes us consider the separation of latent state representation and regularization of the network. Specifically, when we design the network, the latent state representation decided by input data and the regularization by the sparsity constrain need to be taken into account at the same time. The sparsity constrain is realized by adding a term to loss function which will penalize the excessive *activations*, these terms are:

- L1 Regularization: the added term is the absolute value of the vector of *activations* a in layer h

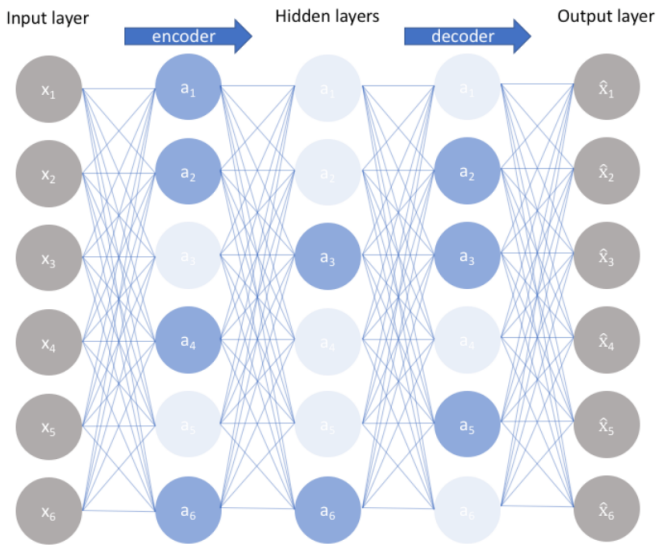


Figure 14: The structure of a Sparse Autoencoder

for input i. The purpose for adding this term is making as many as possible *activations* to zero. This will not influence the representation of latent state while satisfy the sparsity constrain.

$$L(x, \hat{x}) + \lambda \sum_{i} \left| a_i^{(h)} \right| \tag{4.3}$$

- KL-Divergence: the added term is KL-divergence which is a measure of the difference between two probability distributions. Provided that ρ is the average activation of a neuron over a collection of inputs which can be calculated as $\hat{\rho}_j = \frac{1}{m} \sum_{i} \left[a_i^{(h)}(x) \right]$. The subscript j represents the specific neuron in layer h. Each input is denoted as x and m represents the number of inputs. In fact the constrain of activation value of a neuron equals to limit the input to a neuron. If ρ follows Bernoulli distribution, the KL divergence can be used to measure the ideal distribution ρ to the observed distribution over all hidden layer nodes $\hat{\rho}$.

$$L(x, \hat{x}) + \sum_{j} KL(\rho || \hat{\rho}_j) \tag{4.4}$$

The reason for choosing Bernoulli distribution as the neuron activation probability is because the calculation of this value is perfectly match the fire of a neuron. The explain of Bernoulli distribution is "when the probability of a random variable takes value 1 is p, the probability of the variable takes value 0 is $q = 1 - p$".
Actually, ρ represents the percentage of activated neurons in all hidden neurons, it is a value close to 0 (followed Bernoulli distribution). When $\hat{\rho}_j = \rho$, the average activation of all neurons is sparse. KL divergence is a metric to measure the distance between ρ and $\hat{\rho}$, the smaller of

KL divergence the more closer of these two variables which means the more sparse of the network.

4.4 Choice of classifiers

4.4.1 The construction of baseline

In order to compare the performance of deep learning based classifiers for financial fraud detection, we need to build a base model which already have good performance on the dataset. As GBDT is a popular machine learning algorithm which have obtained good results on tabular data in many fields such as intrusion detection system. We decide to use lightGBM as our base model, which is an effective implementation of GBDT, due to its efficiency, accuracy and interpretability. LightGBM is an implementation of GBDT with Gradient-based One-Side Sampling(GOSS) and Exclusive Feature Bundling(EFB). To introduce this model explicitly, we need to understand the theory of GBDT first. Then the improvement method(GOSS and EFB) which used to increase the performance of this model will be explained in detail.

Gradient boosting decision tree(GBDT) uses the linear combination of base model to build the final model. Below is the training process of a GBDT model.

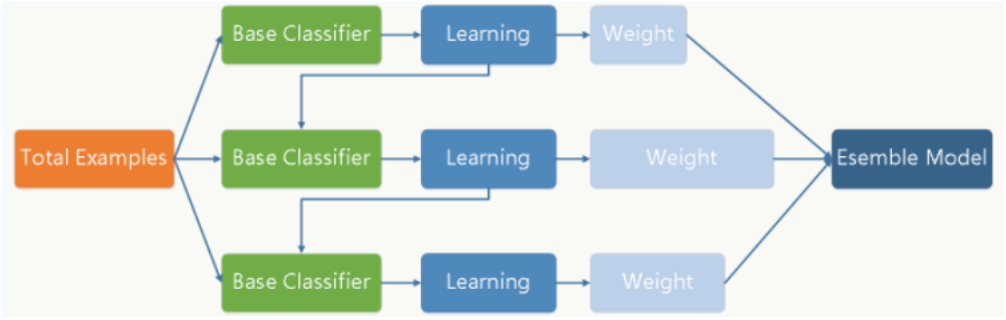


Figure 15: The training process of GBDT model.

GBDT combines multiple weak classifiers to form the final ensemble model. The model training is a multi-iteration process, each iteration generates a weak classifier and the residual of this classifier will be used for training the next classifier in next iteration. The requirement of a weak classifier is that it should have low variance and high bias, and the complexity of the classifier should be simple. The purpose of the ensemble model is by reducing the bias of weak classifiers to increase the accuracy of the ensemble model. Usually, the weak classifier is CART TREE, and due to the requirement of high bias and simplicity, the depth of a tree should not be very large. The ensemble model is linear weighted summation of all weak classifiers.

In the process of one iteration, the classifier learned in preceding iteration is $f_{t-1}(x)$ and the loss function is $L(y, f_{t-1}(x))$. The purpose of this iteration is finding a weak classifier for CART TREE $h_t(x)$ and minimizing the loss function $L(y, f_t(x)) = L(y, f_{t-1}(x) + h_t(x))$. For example, providing that a book is sold for 30$, we use 20$ to fit the price and find that the loss is 10$. Then, we use

6$ to fit the remaining loss and find that there 4$ left to match the target price. In the third round, we use 3$ to find the remaining loss in previous round and find that only 1$ left for the target price. After each iteration, the loss to fit the target will decrease. Method used for fitting loss value is proposed by Freidman[41] which uses negative gradient value of loss function to fit the loss of each iteration. The negative gradient value of loss function for the ith sample in tth iteration can be denoted as:

$$r_{ti} = -[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}]_{f(x)=f_{t-1}(x)} \tag{4.5}$$

Via utilizing $(x_i, r_{ti})(i = 1, 2, \ldots, m)$, the tth CART TREE is obtained. Leafs of this tree can be represented as $R_{tj}, j = 1, 2, \ldots, j$, and j is the number of leafs.

For ever leaf node, we can use all samples in this leaf node to calculate the output value c_{tj} which can minimize the loss function:

$$c_{tj} = \underbrace{\arg\min}_{c} \sum_{x_i \in R_{tj}} L(y_i, f_{t-1}(x_i) + c) \tag{4.6}$$

By adding the output of all leafs, we can get the function used to fit the CART TREE:

$$h_t(x) = \sum_{j=1}^{J} c_{tj} I(x \in R_{tj}) \tag{4.7}$$

The classifier of this iteration can be described as:

$$f_t(x) = f_{t-1}(x) + \sum_{j=1}^{J} c_{tj} I(x \in R_{tj}) \tag{4.8}$$

By using negative gradient value to fit the loss function, we find a method which uses gradient value to fit the bias of loss value. No matter for classification or regression problems, we can use negative gradient value of the loss function to fit the target, the difference simply is the representation of loss function. When the training process finished, a series CART TREES have been built. These trees ensemble together to construct GDBT model.

As said before, LightGBM is a new implementation of GBDT with GOSS and EFB. It overcomes the drawback of GBDT which cannot handle large amount of data and speeds up the training process of conventional GBDT by up to over 20 times while maintaining the same accuracy[29]. The reason of low efficiency for GBDT is that to estimate the information gain of all possible split point, GBDT needs to scan all samples for every feature which is time and resource consuming. LightGBM focus on these problems and devices Gradient-based One-Side Sampling(GOSS) and Exclusive Feature Bundling(EFB) techniques to reduce the sampling amount and the number of features to improve the performance of GBDT.

Gradient-based one-side sampling(GOSS) is a sampling method which can reduce the number of training samples for calculation information gain while keep the accuracy of learned decision trees. The idea of this method is retaining samples with large gradient as they contribute more to information gain and randomly sampling some samples with small gradients. In order to compensate

the influence of data distribution, a multiplier is introduced for data samples with small gradients. Specifically, all samples are sorted according to the gradient value firstly. Then, top $a \times 100\%$ samples with large gradient are selected. Next step is randomly selecting $b \times 100\%$ samples from rest data with small gradients. In order to keep the original distribution of data, a multiplier $\frac{1-a}{b}$ is added to samples with small gradients when calculating information gain. These operations focus on under-trained samples(i.e. samples with large gradients) while do not change too much to the original distribution of data.

Exclusive feature bundling(EFB) commences from the perspective of reducing the number of features when building a decision tree. In most situation, high-dimensional data are sparse and many features are mutually exclusive. The definition of mutual exclusive feature is that they never take nonzero value simultaneously. The exclusive features can be bundled together to form a single feature(called exclusive feature bundle). This operation can reduce the dimension of data, hence the training time can be compressed without hurt the accuracy.

The first step of this method is to determine which features should be bundle together. The idea of graph coloring problem is borrowed for selecting optimal feature bundle. Features are deemed as vertices and edges are added to mutual exclusive features. Greedy algorithm is applied to dye vertices to produce bundles. To further improve the compression result, a small fraction of conflicts between exclusive feature is allowed which means that there is a small possibility that these features take nonzero value simultaneously. The small sacrifice of accuracy can largely improve the computational efficiency. Below is the description of feature bundling algorithm[29]:

Algorithm 1 Greedy Bundling

Input: K: features, K: max conflict count
Output: *bundles*

```
Construct graph G
searchOrder ← G.sortByDegree()
bundles ← , bundlesConflict ←
for i in searchOrder do
    needNew ← True
    for j = 1 to len(bundles) do
        cnt ← ConflictCnt(bundles[j],F[i])
        if cnt + bundlesConflict[i] ≤ K then
            bundles[j].add(F[i]), needNew ← False
            break
        end if
    end for
    if needNew then
        Add F[i] as a new bundle to bundles
    end if
end for
return bundles
```

The algorithm includes three stage: (1) constructing a graph with weighed edges, weight of an

edge represent the conflicting level between two features; (2) features in the graph are sorted in descending order; (3) for every feature in the order list, it can be assigned to an existing bundle, or using it to create a new bundle.

After finding bundles with exclusive features, the second step is merging features in the same bundle together. It is important to ensure that the original value of features can be identified from feature bundle. This can be implemented by adding offset to the original values of features. For example, feature A takes value from [0,10] and feature B takes value from [0,15]. To put these two value together, an offset 10 is added to feature B which making feature B takes value from [10,25]. Thus, feature A and B are bundled together with range[0,25].

LightGBM can significantly outperform XGBoost and GDBT in terms of memory consumption and computational speed. Therefor, we decided to use this powerful model as the baseline which can offer a sound prediction result on the given dataset.

4.4.2 Deep learning based classifiers

With the development of deep learning technique, more and more deep learning topologies are emerged. Each of them has its own characteristic which makes it suitable for a class of problem. For example, Faster recurrent Neural Networks(Faster RNN) are a series networks which obtain remarkable results in image segmentation. In this thesis, we tried two most popular deep learning topologies for financial fraud detection, both of them perform better than base model. In the following part, we will introduce the structure of network used in this thesis.

Convolutional Neural Networks is a classical neural network which has widely used in image recognition area. A convolutional neural network is a stacking of several different layers which includes convolution layer, pooling layer and fully connected layer.

Convolution layer is called **Kernel/Filter, K,** represented in dark blue color. The size of K is a $3 \times 3 \times 1$ matrix. The Kernel shifts with the Stride Length $= 1$, every time performs a matrix multiplication operation between K and the corresponding area of image. The objective of convolution operation is to extract high-level features such as edges of the input image. The shallow layer of the network extract low-level features such as edges, color etc., and the deep layer extract high-level features such as the skeleton of a image. Besides, the size of extracted features can be adjusted by Stride and Padding. Stride is the length of shifting kernel and Padding is pixel which added to the outer edge of an image.

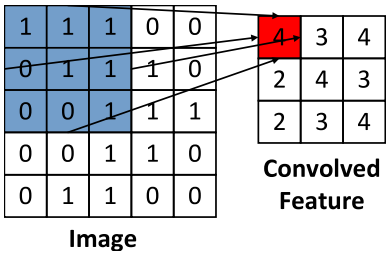


Figure 16: Convolutioning a 5*5*1 image with a 3*3*1 kernel to get a 3*3*1 convolved feature[5]

Pooling layer is usually connected to convolution layer which is responsible for further reducing the size of convolved feature. The function of pooling layers is decreasing computational power by extracting dominant features which are positional and rotational invariant. There are two types of pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value of several patches of convolved feature and average pooling returns the average value of these patches. In most case, we take Max Pooling instead of Average Pooling as it discards noisy at the same time with dimension reduction. Convolutional layer and pooling layer can be stacked many times to form the convolutional neural networks(CNN). The output of them is flattened to be feed to fully connected layer for classification.

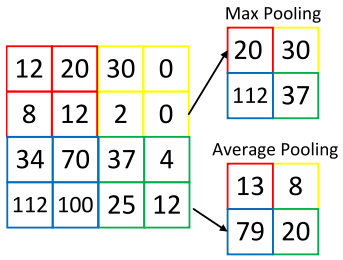


Figure 17: Types of pooling[5]

Fully connected layer learns the non-learner representation of high-level features extracted by convolutional layer. The input to fully connected layer is the perceptron of input image which is then flattened as a column vector. The output of fully connected network is fed to a feed-forward neural network and backpropagation is applied to refine parameters of the network. In order to finish the classification task, a softmax function is applied on the output of fully connected layer.

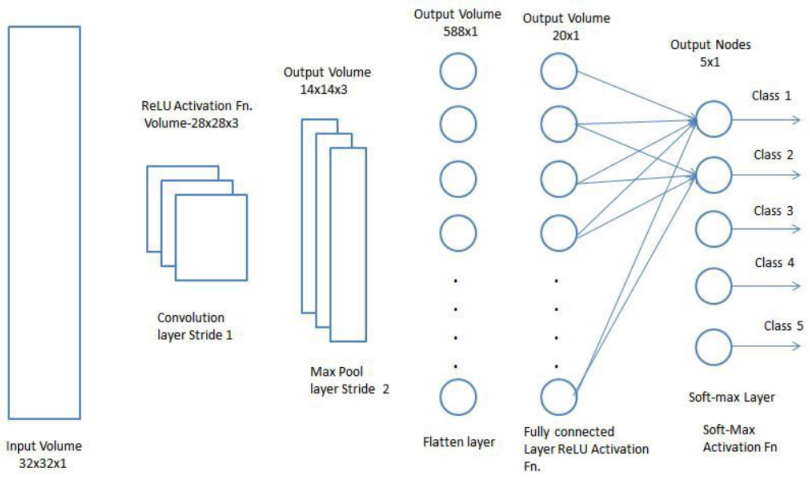


Figure 18: The structure of fully connected layer.[5]

There are many classical CNNs such as AlexNet, VGGNet. Here, we only illustrate the structure of network used in this thesis:

Table 1: The structure of convolutional neural network

Layer(type)	Output Shape
conv2d_1(Conv2D)	(None,15,15,32)
conv2d_2(Conv2D)	(None,15,15,32)
max_pooling2d_1(MaxPooling2)	(None,8,18,32)
conv2d_3(Conv2D)	(None,8,8,64)
conv2d_4(Conv2D)	(None,8,8,64)
max_pooling2d_2(MaxPooling2)	(None,4,4,64)
conv2d_5(Conv2D)	(None,4,4,128)
conv2d_6(Conv2D)	(None,4,4,128)
max_pooling2d_3(MaxPooling2)	(None,2,2,128)
flatten_1(Flatten)	(None,512)
dense_1(Dense)	(None,128)
dropout_1(Dropout)	(None,128)
dense_2(Dense)	(None,128)
dropout_2(Dropout)	(None,128)
dense_3(Dense)	(None,1)
activation_1(Activation)	(None,1)

Long short term memory networks(LSTMs) is a special kind of recurrent neural network. It overcomes the drawback of simple recurrent neural network which is not capable of memorize long term information. Inheriting from recurrent neural networks, LSTM is composed of a chain of repeating modules. Figure 19 gives an image of a single layer for recurrent neural network. From the image, we can see that the structure of a module is pretty interesting which will be explained in detial in the following part.

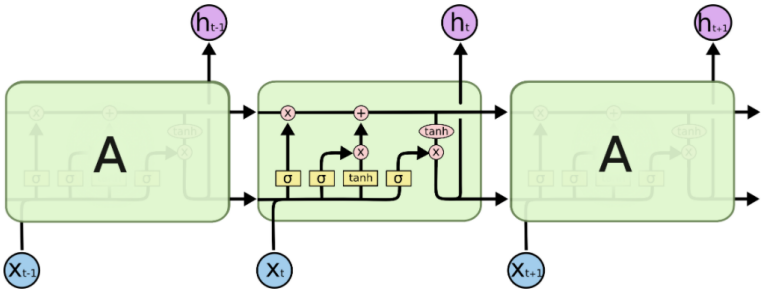


Figure 19: The repeating module in a standard RNN contains a single layer.[6]

The inputs of a module is composed of input from outside and the output of previous module. The output of a module is controlled by different kind of gates. The horizontal line on the top of the diagram runs through the entire chain represents the cell state.

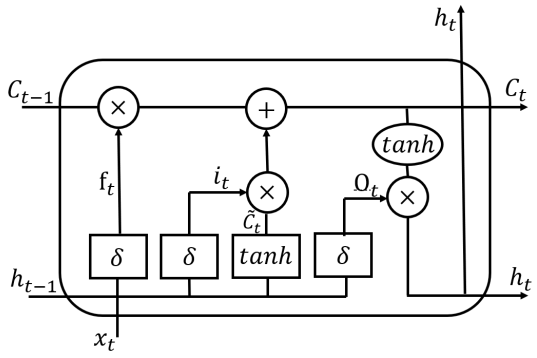


Figure 20: The structure of module

The change of cell state is controlled by gates. A gate consists a sigmoid function with a pointwise multiplication operation. As the output of sigmoid function is between 0 and 1, it decides the output of how much of each component should be let through. A value of 1 means every thing can pass the gate while 0 means no thing is allowed to pass it.

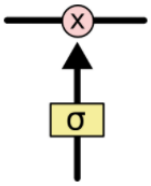


Figure 21: The structure of gates.[6]

The entire operation of a module can be separated into four steps:

1. First step is to decide what information needs to be dropped by cell state, this is decided by sigmoid layer called "forget gate layer". The forget gate takes h_{t-1} and x_t as input, then output a value in the range 0 to 1 which indicates how many proportion of C_{t-1} will be kept.

$$f_t = (W_f[h_{t-1}, x_t] + b_f) \tag{4.9}$$

2. The second step is to decide what information need to store in the cell state. This step includes two part: the first part uses a sigmoid function(called "input gate layer") decides which value need to be updated; the second part uses a tanh function to create a new value $\tilde{c}_t$ which is added to cell state.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4.10}$$

$$\tilde{c}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{4.11}$$

3. The third step is updating the old cell state C_{t-1}. The old cell state C_{t-1} multiplies forgetting gate f_t which decides how much proportion of old cell state need to be thrown away. The remaining part adds the new cell state which is scaled by how much proportion of the new cell state will be used.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4.12}$$

4. The last step is deciding what is going to output. This step also includes two part: the first part uses a sigmoid function to decide what parts of the cell state will be output; the second part uses a tanh function to deal with the cell state and multiply it with the output of sigmoid function. Until now, the output of a module is generated.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4.13}$$

$$h_t = o_t * \tanh(C_t) \tag{4.14}$$

LSTMs use module as neural of a network and stack layers composed by modules to build a network. The structure of LSTM used in our experiments is illustrated in table 2:

Table 2: The structure of long short term memory network

Layer(type)	Output Shape
lstm_1	(None,15,1,64)
dense_1(Dense)	(None,2)

5 Experiments

The chapter present experiment results of the designed methods which tries to explore the answer of proposed questions. It will introduce everything in detail include the experiment environment, dataset and results of experiments.

5.1 Experimental environment

Deep learning is a resource consuming method which need the support of Graphic Processing Unit(GPU). Compared with image and natural language processing, structured data do not need that much resource for storage and computation. Hence, all the experiments is finished on a desktop computer. The equipments of this computer are as follows:

- Processor: Intel Quad Core i7-8700 3.20GHz
- Memory: 32G RAM
- Hard Drive: 480GB
- Operating System: Windows 10 Pro 64-bit
- GPU NVIDIA GeForce GTX 1070
- GPU Memory: 8GB

All codes are implemented in Jupyter written by Python, as for machine learning learning and deep learning method, there are some open library and frame work can be utilized:

- Python 3.7.7
- Sklearn 0.22.1
- Keras 2.3.1

5.2 Dataset

Due to the sensitivity of financial data, it is pretty hard to get real transaction data from financial organizations. Luckily, some organizations would like to share part of their encrypted data to public, hoping to encourage more people to make contribution to this problem. We use two datasets to verify the performance of our proposed method, both of them contain cradit card transaction data. The first dataset comes from a fraud detection competition called IEEE-CIS. It is a typical binary classification competition, the data offered by this competition is pretty good. The amount of bad customer reached to 20000+. The ratio of good to bad customer is around 29 to 1. The dataset includes two tables: transaction table and identity table. As each table has many attributes, the following part will introduce the meaning of every attribute in detail:

Transaction table

TransactionID:primary key.

TransactionDT: timedelta comes from appointed reference time.

TransactionAMT: transaction amount.

ProductCD: product type.

card1 card6: information of payment card, such as card type, issued bank, issued nation etc..

add1 add2: address of payment card.

dist1 dist2: distance.

C1 C14: encrypted count numbers, the real meaning of it is obscured.

D1 D15: encrypted timedelta, for example the time interval between two payments.

M1 M9: match, such as names on card and address, etc..

Vxxx: Vesta engineered rich features , including ranking, counting and other entity relations.

Identity table

Attributes in this table is identity information related to transaction: internet connection information(IP, ISP and proxy etc.) and digital signature(browser version, operating system version etc.). They are collected by Vesta's fraud detection system and data security corporation. Names of attributes are shield, and encryption protocol will not be disclosed to public.

TransactionID: primary key.

Device Type: the type of user's device.

DeviceInfo: the information of user's device.

id1 id38: features related to network connection and digital signature.

The second dataset is a small dataset which contains transactions made by credit card in two days. The are 284,807 transactions in total and only 0.172% of them are frauds. Each sample of this dataset contains 28 encrypted attributes named from 'V1' to 'V28', and two plain attributes 'Time' and 'Amount'. Feature 'Time' is the timestamp for each transaction and feature 'Amount' represents the amount of money transferred by each transaction. Feature 'V1' 'V28' contain transaction information which cannot be disclosed to public. All attributes contained in this dataset are numerical features. Though this dataset is very simple, a lot of academic results are obtained from this dataset[25, 42]. All proposed classification methods are applied on this dataset to recheck the conclusion obtained from the former dataset.

5.3 Experiment results

5.3.1 Features selected by feature engineering

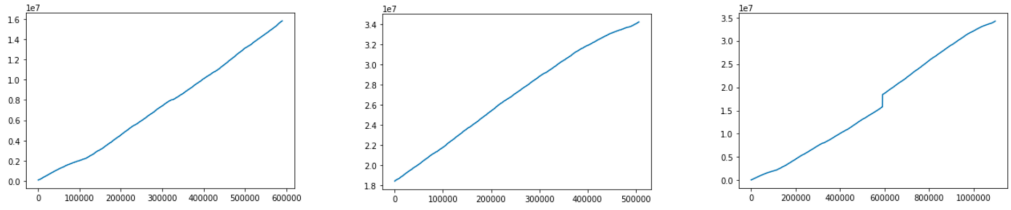
It is important to conduct feature engineering for this dataset. Table 1 contains 394 features and table 2 contains 41 features. There is a lot of redundant information in this feature set, using all features to train a model will decrease the efficiency and accuracy of the model. Due to the amount of features are huge and we need to analyze them one by one, the work of feature engineering is really large. This part only illustrate representative examples of them, and most of them are recorded in jupyter note.

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	card4	card5	...	V330	V331	V332	V333	V334	V335	V33
0	2987000	0	86400	68.5	W	13926	NaN	150.0	discover	142.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal
1	2987001	0	86401	29.0	W	2755	404.0	150.0	mastercard	102.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal
2	2987002	0	86469	59.0	W	4663	490.0	150.0	visa	166.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal
3	2987003	0	86499	50.0	W	18132	567.0	150.0	mastercard	117.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal
4	2987004	0	86506	50.0	H	4497	514.0	150.0	mastercard	102.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.(
5	2987005	0	86510	49.0	W	5937	555.0	150.0	visa	226.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal
6	2987006	0	86522	159.0	W	12308	360.0	150.0	visa	166.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal
7	2987007	0	86529	422.5	W	12695	490.0	150.0	visa	226.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal
8	2987008	0	86535	15.0	H	2803	100.0	150.0	visa	226.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.(
9	2987009	0	86536	117.0	W	17399	111.0	150.0	mastercard	224.0	...	NaN	NaN	NaN	NaN	NaN	NaN	Nal

10 rows × 394 columns

Figure 22: Examples of transaction data in IEEE-CIS.

There are two types data in the table: categorical data and numerical data. To familiar with these data, exploratory data analysis(EDA) is an inevitable process. From figure 23, we can say that transactionDT is a feature which plays the role of timestamp. This feature cannot be used for training a model due to the distribution of this feature offsets seriously between training set and testing set. But this feature still very important as we can use it as x-axis to plot pictures with other features. This will help us to find features which has strong correlation with time.



(a) (b) (c)

Figure 23: Plot of TransactionDT. (a) is the plot of TransactionDT for training set. (b) is the plot of TransactionDT for testing set. (c) is the plot of TransactionDT for merged training and testing set.

For some features, for example TransactionAmt, the distribution of this feature vary a lot in training and testing set, this is not a good sign for selecting this feature. However, the distribution of good and bad transaction samples also vary a lot on this feature. Hence, we can keep this feature as it has strong distinction capability for detecting fraud transaction.

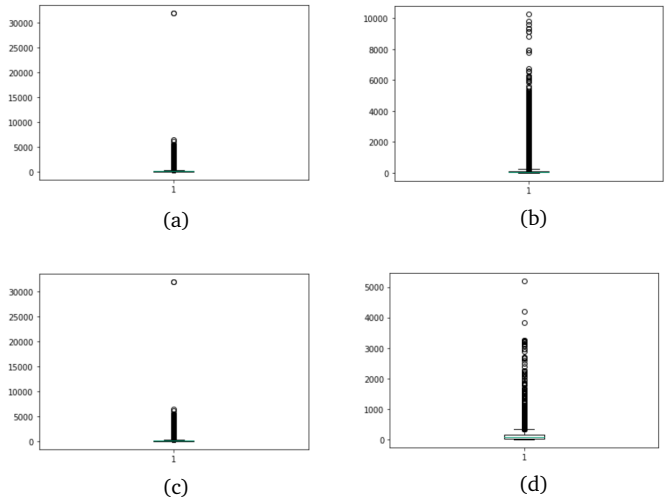


Figure 24: Plot of TransactionAMT. (a) is the distribution of TransactionAMT on training set. (b) is the distribution of TransactionAMT on testing set. (c) is the distribution of TransactionAMT on training set which only contains good samples.(d) is the distribution of TransactionAMT on training set which only contains bad samples.

The ideal features are those have small distribution difference on training and testing set, but have large distribution difference on good and bad samples, for example ProductCD. The distribution of this feature on training and testing dataset is similar while the distribution of good and bad samples on training set vary a lot. From Figure 25(b) we can see that fraud easily happens on product C.

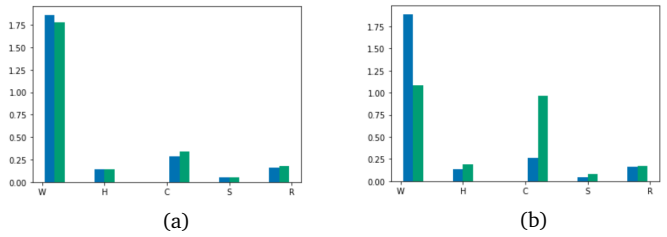


Figure 25: Plot of ProductCD. (a) is the histogram of ProductCD for training and testing data. (a) is the histogram of ProductCD for good and bad samples on training set.

Besides, we need to be clear that not every feature can be plotted, for some numerical features which have many different values, it is hard to plot image for them. Instead, we use statistical analysis to obtain some statistical characteristics of these features. After the exploratory data analysis, we can get a general understanding of all features, this will benefit us a lot for the next feature

processing stage.

After exploring all data, we begin to reduce the amount of features, correlation analysis is used for feature reduction. If two features are highly correlated, i.e. the correlation value between these two features over 0.98, we can drop one of it to reduce the number of features. The difficulty is how to group features together as if we try all possible combination, the computation time will up to $O(n!)$. Here we use a trick that is grouping features which have same amount missing value in the same place together, this will help to reduce the complexity of calculating correlation values. Figures 26 is an example of using heatmap to calculate the correlation values for features: $'TransactionDT', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'D11'$.

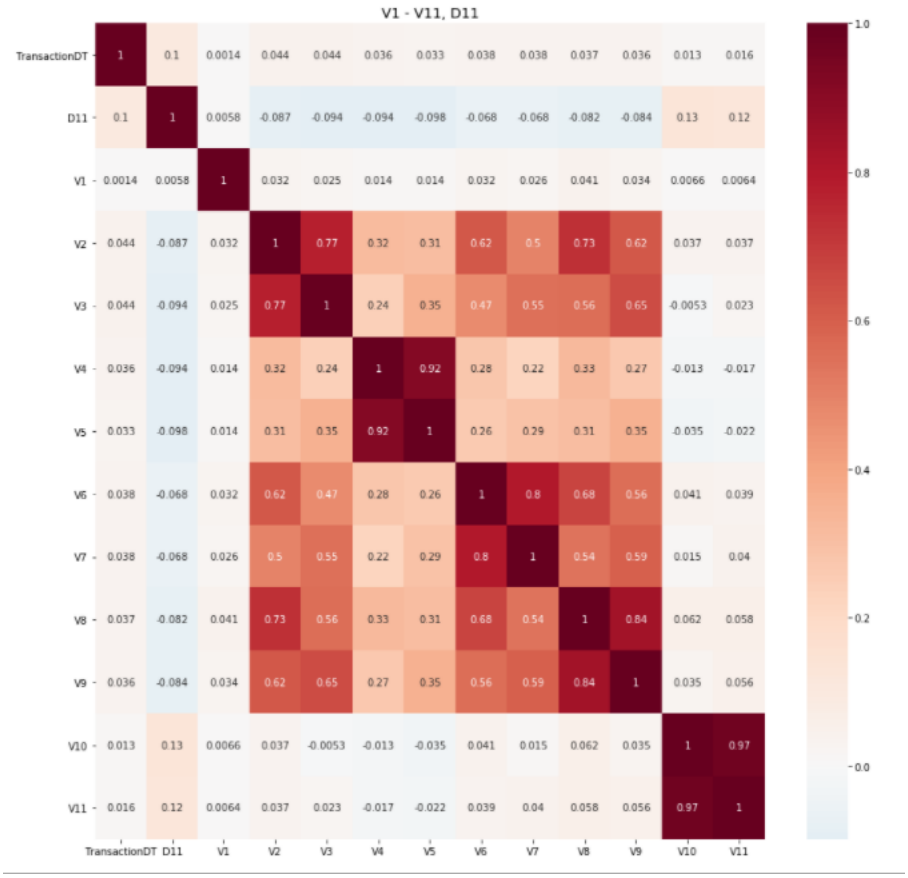


Figure 26: Heatmap for features: TransactionDT, V1 V11 and D1.

After correlation analysis, the feature set for training a model has been decided. The finally feature set contains 213 features which is less than half of original feature set. Though the size of feature set has been minimized, this does not mean that the performance of trained model is good enough. In most case, we need to manipulate features by manual which will further explore

the latent relationship between features. There are many feature derivation techniques and we only use a widely used method which group two or three features together to form new features. To use categorical feature to drive new features, the first step is ordering features according to its importance. We select 7 top most important categorical features to derive new features, the selected features are: $'card1',' card2',' R_emaildomain',' ProductCD',' addr1',' M4',' M6'$.

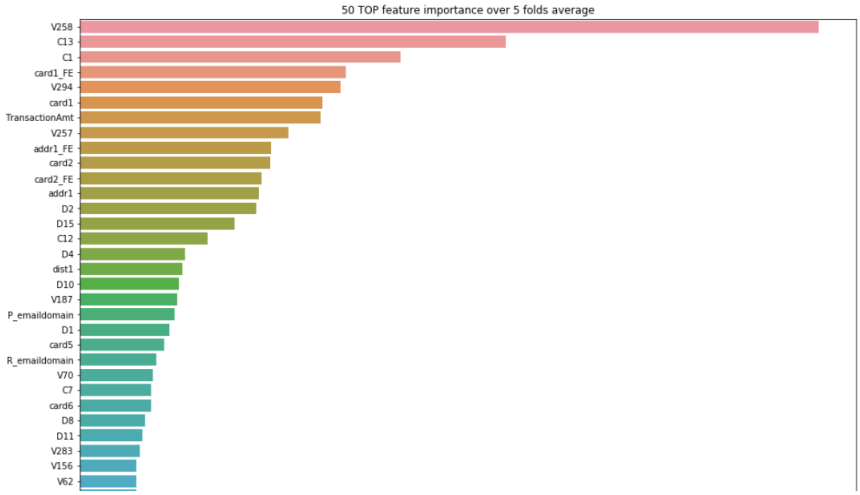


Figure 27: The rank of features by feature importance.

Due to the limitation of memory capacity, we only group two or three features together which results in 56 derived features. The method for selecting derived features can be described as 3 steps:

1. concatenating derived features with original feature set, and calculating the correlation matrix
2. choosing several different threshold for feature selection
3. using feature sets decided by selected threshold to train models, the model with best performance decides the chosen feature set.

We choose three thresholds for feature selection: 0.79999, 0.89999 and 0.98999. Using feature sets selected by these thresholds to train models separately. The classification model used in our experiments is LightGBM with KFold cross validation. According to experiment results, 0.989999 is the best threshold for feature selection. Actually, in most case, the threshold for feature selection should not be very small as this may cause many information loss. The final derived features we decided to use are: $'ProductCD\_M4',' ProductCD\_M6',' M4\_M6',' card1\_R\_emaildomain\_addr1'$, $'card2\_R\_emaildomain\_addr1',' R\_emaildomain\_ProductCD\_addr1',' ProductCD\_addr1\_M4'$, $'ProductCD\_M4\_M6',' addr1\_M4\_M6'$.

Table 3: The performance of LightGBM with different feature sets decided by different thresholds

Threshold	0.79999	0.89999	0.98999
Avg accuracy	0.935387	0.935786	0.935580

After the complex feature engineering process, the entire feature set used for training a model is decided which contains 222 features. Next step is using a deep learning approach to solve dataset imbalance problem.

5.3.2 The analysis on approaches for dataset imbalance

The dataset for fraud detection is seriously imbalanced, this will influence the prediction accuracy. The idea to solve this problem is downsampling and oversampling which try to balance the amount of samples in different dataset. In this thesis, we compare four strategies to balance the amount of data in different dataset.

- Random down sampling samples from the majority class to balance the amount of data in two dataset.
- Over sampling samples in minority class by SMOTE algorithm. Due to the data amount is too large, if we over sample samples in minority class to the same amount of majority class, it will consume too much computing time and resource. Hence, we only over sample samples in minority class to half amount of majority class and down sample samples in majority class to half of it.
- Over sampling samples in minority class by Autoencoder. Similar to SMOTE method, we only use Autoencoder to over sample samples to half amount of the majority class and down sample samples in majority class to the same amount. To further analyzing the performance of autoencoder, we use two different shape Autoencoder(normal and sparse Autoencoder) to conduct the over sampling task.

The experiment results are showed in table 4. We use LightGBM with KFold cross validation as the baseline to train models with different dataset. According to the experiment results, the prediction accuracy of trained model increased a lot by balancing the dataset. The performance of oversampling method performs much better than down sampling method. However, no matter SMOTE or Autoencoder, the average accuracy of these oversampling methods are over 0.99 which implies the potential of overfitting. In our experiments, the performance of normal autoencoder and sparse autoencoder close to each other which means that the topology of autoencoder has not that large impact on synthesizing samples. The prediction accuracy of SMOTE and proposed autoencoders are similar, hence we can say that the proposed autoencoders for oversampling perform well on solving dataset imbalance problem.

Table 4: The comparison of sampling strategies for imbalanced dataset

Sampling Method	Avg accuracy
baseline	0.935625
DownSampling	0.960215
SMOTE	0.998554
Autoencoder(normal)	0.998075
Autoencoder(sparse)	0.998038

As the input to autoencoder is reshaped as image-like form, which is of size 15 multiply 15, the input and output of an autoencoder can be visualized by image. Figure 28 plots 10 samples' input and output images. From the images, we can see the difference between inputs and outputs. Although the performance of model trained on the augment dataset is not bad, the difference between real sample and synthesized sample is still very large.

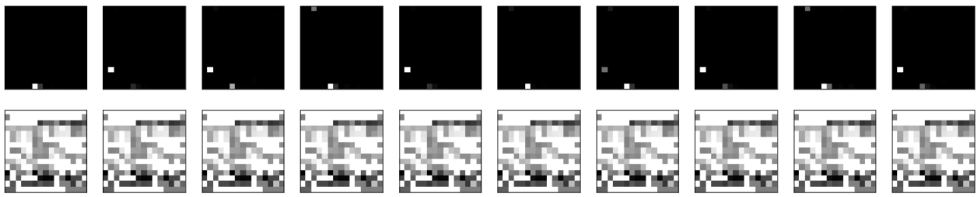


Figure 28: Visualization inputs and outputs of Autoencoder

Though the output of an autoencoder looks not that similar to the original sample, it has strong positive correlation with the label of input sample. Figure 29 illustrates the correlation between synthesized samples and the label of input samples. From the picture, we can draw the conclusion that though the synthesized samples looks not that similar to the input samples, but they inherit the key features of original samples which benefits for classification task.

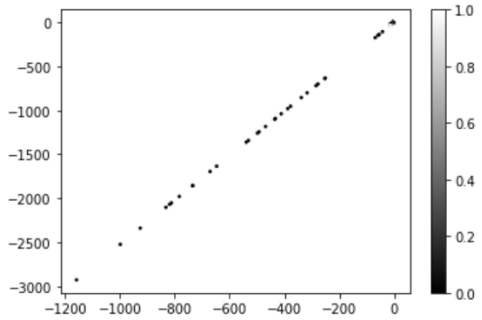


Figure 29: The correlation between synthesized samples and the label of input samples

5.3.3 The comparison of different classifiers

To evaluate the performance of deep learning based classifiers for financial fraud detection, we build a baseline which has been proved perform well in anti-fraud field. The base model chosen in this thesis is LightGBM which is an improved version of gradient boosting decision tree. The choice of cross validation method influences the performance of trained model. We use four different cross validation methods(StratifiedKFold, KFold, TimeSeriesSplit, GroupKFold) to train models separately, and select the best performance method as our cross validation method for LightGBM. According to our experiments, KFold and GroupKFold obtain the best performance, the average accuracy reach to 0.928. Hence, we decide to use KFold cross validation method to train LightGBM. Here we give a simple explain on KFold cross validation, training set is splitted into k folds, the model is trained for k round with 1 fold left out for validation each round.

Table 5: The comparison of different cross validation strategies

Cross Validation method	Avg accuracy
StratifiedKFold	0.914327
KFold	0.928300
TimeSeriesSplit	0.914021
GroupKFold	0.928637

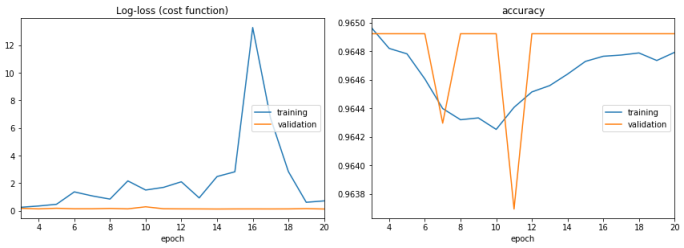
The deep learning models we selected are CNN and LSTM, the structure of these models are described in Chapter 4. According to experiment results, CNN and LSTM outperform LightGBM in predicting accuracy which verifies the power of deep learning models. The prediction accuracy of CNN and LSTM are similar to each other, both of them reached 0.969, this is caused by two reasons: (1) the learning capability of deep neural network is super strong, both of them have extracted sufficient information for the data. (2) the data is not enough for training a robust network. As we all know, deep learning was used to solve image processing problem which need more than 10GB data for training, hence tabular data offered in our experiment is insufficient. Based on above reasons, our networks have possibility of overfitting.

The feature set we used contains 222 features and 9 of them are derived features which manufactured by us. All of these models have been trained on feature set with and without derived features. It is obvious that the performance of LightGBM is seriously influenced by derived features, but for CNN and LSTM, adding derived features almost have no influence on them. The power of deep neuron network to extract high level features from raw data is verified in tabular data field, this can help to simplify the complex feature engineering process in fraud detection field.

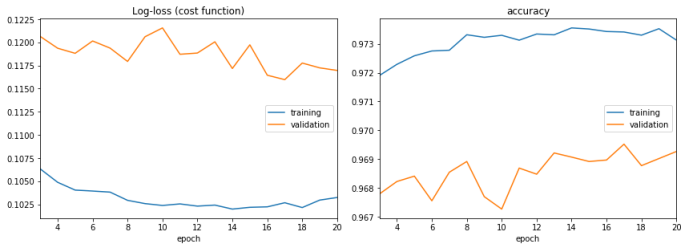
Table 6: The comparison of different classifiers with and without derived features

Classifier	Avg accuracy
LightGBM(without derived features)	0.934343
LightGBM	0.935625
CNN(without derived features)	0.966
CNN	0.965
LSTM(without derived features)	0.969
LSTM	0.969

Although the predict accuracy of CNN and LSTM in final epoch are similar, the performance of the two models are different. Figure 30 illustrates the loss curve and accuracy curve of CNN and LSTM, it is obvious that LSTM is more stable than CNN according to the training records. This is understandable as there are some features have strong correlation with time, LSTM is sensitive to time related features making this model fit better to our data.



(a)



(b)

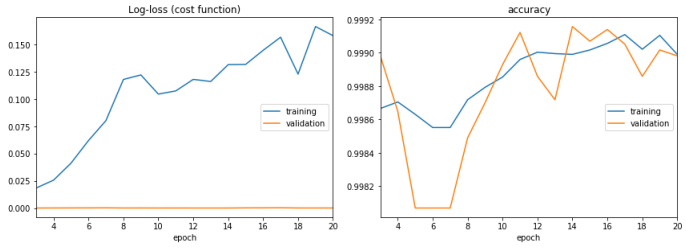
Figure 30: Loss curve and accuracy curve for CNN and LSTM. (a) plots the loss curve and accuracy curve of CNN. (b) plots the loss curve and accuracy curve of LSTM.

As said before, we apply classification methods, i.e. LightGBM, CNN and LSTM on the second credit card fraud detection dataset as well. The base model lightGBM has obtained a fairly good result which the AUC value is 0.984239. While the prediction accuracy of deep learning based models close to 1 which are still better than normal machine learning method. This experiment result supports our conclusion that deep learning based models perform better than traditional machine learning tools for fraud detection problem.

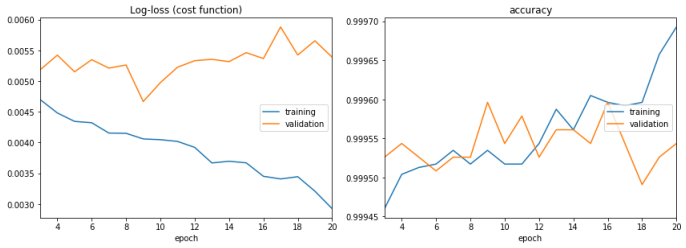
Table 7: The comparison of different classifiers on second credit card fraud detection dataset

Classifier	Avg accuracy
LightGBM	0.984239
CNN	0.999
LSTM	1

Though we have also removed the attribute 'Time' from this credit card fraud detection dataset, there should be some time related features among the encrypted features as LSTM still performs better than CNN on this dataset. The prediction accuracy of LSTM closes to 1 on this dataset and the training loss of this model is 0.003. This amazing result proves the excellent classification capability of LSTM for financial fraud detection problem.



(a)



(b)

Figure 31: Loss curve and accuracy curve for CNN and LSTM on second dataset. (a) plots the loss curve and accuracy curve of CNN. (b) plots the loss curve and accuracy curve of LSTM.

6 Discussion, conclusion and future work

This chapter will answer research questions proposed at the beginning of this thesis. Concluding results of our research and pointing out research direction in future.

6.1 Discussion

This thesis conduct a comprehensive analysis on financial fraud detection from several aspects: inherent features existed in this type data, traditional methods for financial fraud detection and comparison between deep learning based approaches with classical methods. Based on our research, we think deep neuron networks work in this field which deserve more investment in this direction. Here, we will answer questions proposed before:

How to use deep learning technique to solve the dataset imbalance problem of financial fraud data?

In real business activity, fraud behaviors are much less than normal behaviors, hence dataset imbalance is the inherent characteristic of financial fraud data. There are many strategies for confronting this problem, the simplest idea is balancing the data amount of different class. There are two sampling methods: oversampling and downsampling. Downsampling is randomly selecting same amount of data in minority class from majority class. Oversampling is synthesizing samples in minority class until the amount of samples in different classes reach balance. We propose an oversampling method which is based on a kind of deep neuron network called autoencoder. It first encodes input sample into a compressed code, and then decode it to an output which has same shape with input samples. This process repeats many times until the loss between input and output lower than a threshold. We compare the propose oversampling method with another oversampling method called SMOTE, and randomly downsampling method, the proposed method outperforms other methods which answers the question we proposed before. Deep learning technique is a powerful tool for solving dataset imbalance problem, the synthesized data generated by deep neural network augment the dataset which finally improve the performance of trained models.

Can deep learning based classifier perform better than other machine learning methods?

Compared with image and language data, tabular data have already had a series mature techniques to deal with. Linear regression and GBDT are dominant algorithms in these field. The reason for the prevalence of these algorithms are: (1) both of them have obtained good results on financial data; (2) they are simple enough to follow the data flow in the algorithm which makes it possible to monitor every variable in the algorithm function. We compare two types deep neuron network with LightGBM(an improved gradient boosting decision tree), and find that both of these deep neuron networks outperform LightGBM. The structure of deep neuron network enables it to extract rich information from tabular data which makes deep neuron networks more powerful than other ma-

chine learning techniques. Besides, we find that LSTM performs better than CNN on training data, this may be caused by features which have strong correlation with time. LSTM is a deep learning model which is suitable for time series problem, it deeply mines time related features of data and draws smoothly increased training curves.

Does deep learning based classifier simplify the process of feature engineering?

For tabular data in financial field, the size of feature set is very large, feature engineering is an inevitable process in this field. The feature engineering methods used in this thesis includes: exploratory data analysis(EDA), data imputation, correlation analysis and feature derivation. Exploratory data analysis gives us an general image of data which helps research in later stage. Data imputation is a process which includes filling missing values, normalizing the range of values etc., these operations prepare data in good state for building models. Correlation analysis is also an important step in our research work as it can remove redundant features which consume computing time and energy. Feature derivation is a step which utilizes original features to manufacture new features. The purpose of this step is mining the hidden relationship between features to enhance the performance of models. We compare the performance of LightGBM with proposed deep neuron networks, each of them is trained with feature set with and without derived features separately, and obtain the conclusion that feature engineering is important for traditional machine learning models but has less influence on deep neuron networks. The deep and wide network structure is able to extract the hidden relationship between features automatically which helps to simplify the process of feature engineering.

6.2 Conclusion

This thesis conducts a comprehensive analysis on financial fraud detection which undertakes from analysis on financial fraud detection to countermeasures for problems existed in this field. Through literature review, we find that problems existed in this field can be classified in three types: (1) feature sets are large and complex which need data preprocessing and feature engineering to increase the performance of trained model. (2) Due to the happens of fraud behavior is seldom, the dataset of fraud detection is severely imbalanced. It is important to solve dataset imbalance problem before starting detecting fraud. (3) To balance the catching rate of fraud and the experience of customers without too much interception, researchers and engineers are always looking for good classification models.

This thesis focus on problems existed in financial fraud detection field and the state-of-art deep learning techniques, proposed a set of approaches for solving these problems. We first conduct feature engineering on original dataset which includes: exploratory data analysis(EDA), data preprocession, correlation analysis and feature derivation. After this step, the size of feature set reduced to half of its original size while the prediction accuracy of the base model reaches 0.9356. Then, we compare the proposed deep learning based oversampling method(Autoencoder) with classical oversampling method SMOTE and randomly dowm sampling method. All of these sampling methods release the burden of dataset imbalance and largely improve the performance of base model. According to our experiment results, oversampling methods (SMOTE and autoencoder) outperform

downsampling method which indicates that down sampling method may loss some information of majority class. But this does not mean that oversampling methods are hundred percent good to dataset imbalance problem, as they may cause classifiers overfit the training set and decrease the prediction accuracy on testing set. Finally, we compare two deep learning based classifiers(CNN and LSTM) with base model(LightGBM), both of the proposed classifiers perform better than base model(LightGBM) which proves the potentiality of deep learning based classifier on financial fraud detection. Besides, all of these classification models are trained on dataset with derived and underived features separately. Only LightGBM is influenced obviously by derived features, the performance of deep learning based classifiers have similar prediction accuracy no matter the dataset contains derived features or not. This is owed to the complex structure of the network which can extract rich information from inputs automatically, hence deep neuron networks simplify the procedures of feature engineering. However, such many of neurons may cause the problem of overfitting and we need to use other techniques, such as dropout, to avoid such problem.

Though there are still a lot of problems existed in using deep learning based approaches for financial fraud detection. We still hold positive view on applying this technique on financial fraud detection as it is a powerful tool which not only obtains good performance in fraud detection but also can be used for different tasks in this field.

6.3 Future work

This thesis just kick the start of applying deep neuron networks on financial fraud detection. In our future work, we will try to use different topology of deep neuron network for financial fraud detection. Though we only tried two different deep neuron networks and their prediction accuracy is similar, we still find that LSTM performs a little better than CNN due to feature set contains features which are highly related to time. Hence, we believe that the influence of network topology on fraud detection is an interesting research point. We will focus on two aspects: (1) how does different network topologies influence the performance of fraud detection. (2) How does width and depth of the network influence the performance of fraud detection.

There is another hot research topic which relates to online real-time fraud detection. In recent year, online payment becomes a common transaction method, this payment method requires the system detect fraud behavior from paramount data in short time. Most traditional researches focus on offline fraud detection algorithms which is not suitable for real-time fraud detection. Besides, online payments introduce more complex data structure, such as online forms filled by customers, it requires more powerful tools to deal with different data structure at the same time. Luckily, with the emerging of big data techniques and deep learning algorithms, we see the possibility of solving such problem and will conduct analysis in this field lately.

Bibliography

[1] Jain, A. K., Duin, R. P. W., & Mao, J. 2000. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1), 4–37.

[2] Kononenko, I. & Kukar, M. 2007. *Machine learning and data mining*. Horwood Publishing.

[3] Yan, L., Yoshua, B., & Geoffrey, H. 2015. Deep learning. *nature*, 521(7553), 436–444.

[4] Douzas, G., Bacao, F., & Last, F. 2018. Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, 465, 1–20.

[5] Saha, S. A comprehensive guide to convolutional neural networks. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd

[6] colah. Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[7] Aleskerov, E., Freisleben, B., & Rao, B. 1997. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)*, 220–226. IEEE.

[8] of Certified Fraud Examiners, A. 2016. *Report to the nations on occupational fraud and abuse: 2016 global fraud study*. Association of Certified Fraud Examiners.

[9] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. 2018. *Learning from imbalanced data sets*. Springer.

[10] Holte, R. C., Acker, L., Porter, B. W., et al. 1989. Concept learning and the problem of small disjuncts. In *IJCAI*, volume 89, 813–818. Citeseer.

[11] Gao, Y. 2017. Design and implementation of online fraud detection algorithm. http://cdmd.cnki.com.cn/Article/CDMD-10335-1017066842.htm. Accessed: 2017-02-04.

[12] Elkan, C. 2001. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, 973–978. Lawrence Erlbaum Associates Ltd.

[13] Acfe:report to the nations on occupational fraud and abuse, the 2020 global fraud study.

[14] Weisberg, H. I. & Derrig, R. A. 1991. Fraud and automobile insurance: A report on bodily injury liability claims in massachusetts. *Journal of Insurance Regulation*, 9(4).

[15] He, K., Zhang, X., Ren, S., & Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[16] Collobert, R. & Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167.

[17] Bănărescu, A. 2015. Detecting and preventing fraud with data analytics. *Procedia economics and finance*, 32, 1827–1836.

[18] Rushin, G., Stancil, C., Sun, M., Adams, S., & Beling, P. 2017. Horse race analysis in credit card fraud—deep learning, logistic regression, and gradient boosted tree. In *2017 systems and information engineering design symposium (SIEDS)*, 117–121. IEEE.

[19] Schreyer, M., Sattarov, T., Borth, D., Dengel, A., & Reimer, B. 2017. Detection of anomalies in large scale accounting data using deep autoencoder networks. *arXiv: Learning*.

[20] Wang, Y. & Xu, W. 2018. Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud. *Decision Support Systems*, 105, 87–95.

[21] Wang, S., Liu, C., Gao, X., Qu, H., & Xu, W. 2017. Session-based fraud detection in online e-commerce transactions using recurrent neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 241–252. Springer.

[22] Barkan, O. & Koenigstein, N. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. IEEE.

[23] Chouiekh, A. & Haj, E. H. I. E. 2018. Convnets for fraud detection analysis. *Procedia Computer Science*, 127, 133–138.

[24] Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. 2018. Deep learning detecting fraud in credit card transactions. In *2018 Systems and Information Engineering Design Symposium (SIEDS)*, 129–134. IEEE.

[25] Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. 2019. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 479, 448–455.

[26] Spark, A. 2018. Apache spark. *Retrieved January*, 17, 2018.

[27] Spark. Spark streaming. http://spark.apache.org/docs/2.2.0/streaming-programming-guide.html.

[28] Kazemi, Z. & Zarrabi, H. 2017. Using deep networks for fraud detection in the credit card transactions. In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, 0630–0633. IEEE.

[29] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, 3146–3154.

[30] Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. 2015. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, 159–166. IEEE.

[31] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. 2014. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, 41(10), 4915–4928.

[32] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. 2017. Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE transactions on neural networks and learning systems*, 29(8), 3784–3797.

[33] Dal Pozzolo, A. 2015. Adaptive machine learning for credit card fraud detection.

[34] Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. 2018. Scarff: a scalable framework for streaming credit card fraud detection with spark. *Information fusion*, 41, 182–194.

[35] Carcillo, F., Le Borgne, Y.-A., Caelen, O., & Bontempi, G. 2018. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization. *International Journal of Data Science and Analytics*, 5(4), 285–300.

[36] Lebichot, B., Le Borgne, Y.-A., He-Guelton, L., Oblé, F., & Bontempi, G. 2019. Deep-learning domain adaptation techniques for credit cards fraud detection. In *INNS Big Data and Deep Learning conference*, 78–88. Springer.

[37] Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. 2019. Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*.

[38] Zheng, A. & Casari, A. 2018. *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.".

[39] Wang, B. X. & Japkowicz, N. 2004. Imbalanced data set learning with synthetic samples. In *Proc. IRIS Machine Learning Workshop*, volume 19. sn.

[40] Jordan, J. Introduction to autoencoders. https://www.jeremyjordan.me/autoencoders/.

[41] Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

[42] Alghofaili, Y., Albattah, A., & Rassam, M. A. 2020. A financial fraud detection model based on lstm deep learning technique. *Journal of Applied Security Research*, 15(4), 498–516.