Undergraduate Project Report
2021/22

A Study of Feature Selection Methods for Classification

Name:	Chenyu Liu
School:	International School
Class:	2018215115
QMUL Student No.:	190016069
BUPT Student No.:	2018213029
Programme:	e-Commerce Engineering with Law
Supervisor	Dr. Addisson Salazar Prof. Luis Vergara

Date: 28-04-2022

Table of Contents

Abstract

Classification is one of the important tasks of machine learning, which classifies each object in a dataset into its corresponding class based on its features. However, an object might have many features, which leads to many problems that hinder the performance of machine learning algorithms, for example, the curse of dimensionality and overfitting. Therefore, reducing data dimensionality is considered an important approach to dealing with high-dimensional data, and one of the methods to reduce data dimensionality is feature selection, which selects a subset from the entire feature set to maximize the performance of the machine learning algorithms and minimize the number of features. In this project, we implemented a feature selection framework that consists of four parts: data pre-processing, feature selection, classification, and evaluation. Based on this framework, in order to compare the 3 different types of feature selection methods, which are the Filter, Wrapper, and Embedded methods. We combine five feature selection methods and five classification methods on three different datasets and evaluate their performance and computational cost based on the cross-validation split strategy. Through the experiments, we find that Filter approaches are fast and easy to compute, and the Wrapper approach considers the correlations between feature selection and classifier. For Embedded approaches, it combines the common advantages of both methods explained above. Among them, we also propose a Transformer-based feature selection method FS-Former, and we demonstrate through experimental results that our proposed method achieves comparable performance with other feature selection methods.

摘要

分类是机器学习的重要任务之一，它根据数据集中的每个对象的特征将其分为相应的类别。然而，一个对象有许多特征，这将导致许多问题，阻碍机器学习算法的性能，例如，维度的诅咒和过度拟合。因此，降低数据维度被认为是处理高维数据的一个重要方法，而降低数据维度的方法之一就是特征选择，它从整个特征中选择一个子集，使机器学习算法的性能最大化，特征的数量最小化。在这个项目中，我们实现了一个特征选择的框架，这个框架包括四个部分：数据预处理, 特征选择, 分类，评估。基于这个框架，为了比较过滤法（Filter approach），包裹法（Wrapper approach）和嵌入式法（Embedded approach）这三种不同的特征选择方法，我们于三种不同的数据集上组合 5 个特征选择方法和 5 个分类方法，并基于交叉验证法的策略评估他们的性能和计算花费。通过实验，我们发现过滤法是快速和容易计算的，而包裹法考虑了特征选择和分类器之间的关联性。对于嵌入法，它结合了上述两种特征选择方法的共同优点。其中，我们还提出了一个基于 Transformer 特征选择方法 FS-Former，通过实验结果证明，我们所提出的方法达到了和其他特征选择方法可比较的性能。

Chapter 1: Introduction

1.1 Motivation

In recent years, with the rapid development of information technology, machine learning techniques have become an important tool for processing big data. Among them, the classification task is one of the most dominant machine learning tasks which classifies each object in the data set into corresponding classes or categories based on its features. Currently, machine learning classification is mainly used in various fields such as computer-aided diagnostics [1], facial recognition [2], and spam detection [3]. However, with the improvement of data collection technology, the dimensionality of the collected data is also rising, even up to tens of thousands of dimensions (such as biomedical data [4]), which makes it impractical to adopt traditional machine learning techniques to High-dimension low-sample size (HDLSS) data. The problems that hinder the performance of the machine learning algorithm in HDLSS data include the following.

There are two main reasons for the weak generalization ability of machine learning algorithms in high-dimensional data: (1) Curse of Dimensionality: In high-dimensional feature space, the distribution of data is highly nonlinear, and it is difficult to build a suitable interface, which leads to the inability to build classification models with strong generalization ability [5]. (2) In high-dimensional data, the number of samples is relatively insufficient compared with the number of features, and it is easy to make the learning objectives not related to the original This can lead to the failure of machine learning modeling based on empirical data, resulting in poor generalization ability [6].

The root cause of the above problem is that the high-dimensional data in a dataset usually contains a large number of irrelevant and redundant features. Therefore, reducing data dimension is considered to be an essential step in handling high-dimensional data and one of the methods to reduce data dimension is feature selection, which selects a subset of the whole features to maximize the performance and minimize the number of features of machine learning algorithm [7].

Introducing feature selection for machine learning algorithms can have many benefits: (1) It can effectively solve the problem of weak generalization ability caused by the curse of dimensionality and overfitting problems. (2) Due to the reduced number of features in the data, the computational complexity of the machine learning algorithm is also reduced, thus

improving the processing performance of the model. (3) Since only relevant features are selected, it can help researchers to uncover task-relevant features. For example, in DNA gene analysis, feature selection can help researchers to find potential gene expression [8][9][10].

However, the feature selection method aims to find an optimal subset of features from 2^n possible combinations, where n is the number of features. Therefore, it is an NP-hard problem that is very difficult to deal with. To better understand the performance of feature selection methods, in this work, a comparison of 3 main types of feature selection methods is made including Filter approaches, Wrapper approaches, and Embedded approaches, which are classified depending on the different ways of combining machine learning methods and feature selection methods [11]. In the comparison, 3 different datasets and 5 different classification methods are used to better explore the characteristics of different feature selection methods. In addition to the above methods, we also propose an embedded feature selection method, FSFormer, based on Transformer, one of the most popular and state-of-the-art deep learning architectures [12].

1.2 Overview

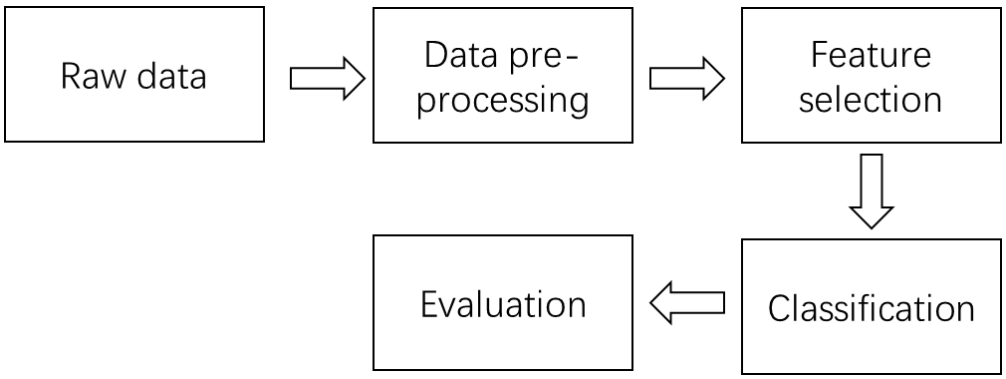


Figure 1 The workflow of our project

As shown in Figure 1, this project is divided into four parts: data pre-processing, feature selection, classification, and evaluation. Here, this report will illustrate these parts briefly.

(1) Data pre-processing: Since the raw data are typically unclean and feeding the raw data into a machine learning model may cause many problems including incompleteness, noise, inconsistency, redundancy, imbalance, outliers, and duplication Thus, an appropriate data pre-processing pipeline is needed to overcome these problems.

(2) Feature selection: This is the main part of our project. The feature selection methods will select the optimal set of features based on specific criteria. Thus, the data are processed to contain only the optimal set of features.

(3) Classification: In this step, the machine learning algorithm will classify each observation using optimal selected features into different categorical values. In this project, we have implemented the following classifiers: linear discriminant analysis (LDA) [13], quadratic discriminant analysis (QDA) [13], Random Forest [14], support vector machine (SVM) [15], and multilayer perceptron (MLP).

(4) Evaluation: To better understand the performance of different feature selection methods, the quality of classification results will be evaluated using several indices such as accuracy, balanced accuracy, confusion matrix, sensitivity, specificity, and Area Under Curve (AUC). Besides, the computational cost of the different cases of classification will also be estimated.

1.3 Contribution

In summary, our main contributions are as follows:

(1) Implementation of the appropriate and effective data pre-processing pipeline for uncleaned raw data.

(2) Implementation of three different types of feature selection methods including Filter approaches, Wrapper approaches, and Embedded approaches.

(3) Implementation of five different widely used classification methods.

(4) Proposal of an embedded feature selection method which is called FSFormer based on the state-of-the-art Transformer architecture.

(5) Adapt and combine the implemented feature selection and classification methods to process three different datasets.

(6) Evaluate the performance of different combinations of the methods in terms of accuracy, balanced accuracy, confusion matrix, sensitivity, specificity, Area Under Curve (AUC), and computational cost.

Chapter 2: Background

2.1 Feature Selection

Feature selection, which is also known as attribute selection, aims to select the optimal subset of features that can maximize the performance and minimize the number of features of the machine learning algorithm. Therefore, this part of the report will introduce some basic definitions of feature selection methods.

2.1.1 Types of features

According to the previous research [16], features in an observation can be divided into three main types: relevant features, irrelevant features, and redundant features. The definitions are as follows:

Here we define the set containing all features as U, the ith features as F_i, and the target information H.

(1) Relevant features: It is helpful for the machine learning algorithm and can improve the performance of the algorithm. If it is eliminated from the feature set, the machine learning performance will be deteriorated, which can be mathematically represented as:

$$P(H \mid U) \neq P(H \mid U - Fi) \tag{1}$$

(2) Irrelevant features: It is not helpful for the machine learning algorithm and will not bring any improvement to the algorithm performance. If it is eliminated from the feature set, the machine learning performance will not be deteriorated, which can be mathematically represented as:

$$P(H \mid U) = P(H \mid U - Fi) \tag{2}$$

(3) Redundant features: The features that can be inferred from existing features, therefore, It doesn't bring any new information to the machine learning algorithm. If it is eliminated from the full feature set, the machine learning performance will not be deteriorated, while if its relevant features are also eliminated, the machine learning performance will be deteriorated. This can be mathematically represented as:

$$P(H \mid U) = P(H \mid U - Fi) \tag{3}$$

$$P(H \mid U - Fj) \neq P(H \mid U - Fj - Fi) \tag{4}$$

where *Fj* is any feature that $j \neq i$.

Therefore, as seen from the definition of features, the purpose of feature selection is to remove irrelevant and redundant features while retaining the relevant ones, and thus find a subset of features that can lead to the optimal performance of the machine learning model.

2.1.2 Process of feature selection

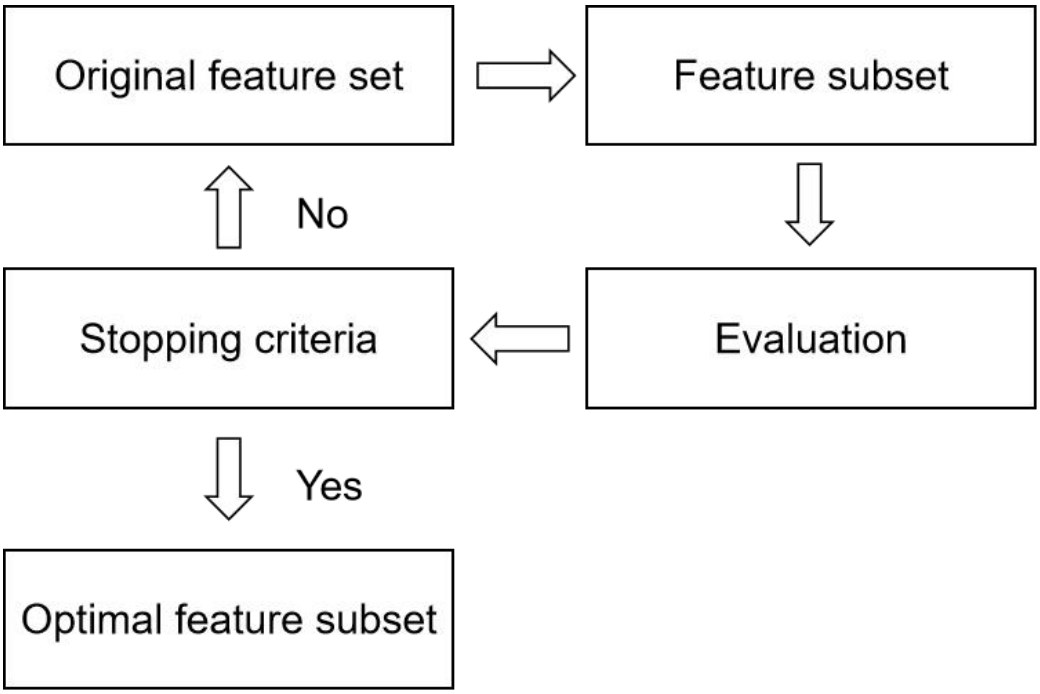


Figure 2 Process of feature selection

The process of feature selection is shown in Figure 2, it first selects a subset of features based on a specific method, then evaluates the selected subset based on specific evaluation criteria, after that, The evaluation result is compared with the stopping criteria of the feature selection process, and if the stopping criteria are satisfied, the final optimal feature subset is obtained, if not, it will continue selecting features until the stopping criteria are satisfied.

2.1.3 Types of feature selection methods

Depending on the different ways of combining feature selection and machine learning algorithms, feature selection methods can be divided into three approaches: Filter approaches,

Wrapper approaches, and Embedded approaches. Here this paper will discuss the characteristics of each method.

(1) Filter approach

It is characterized by the fact that it selects features without evaluating the performance of the machine learning model, that is, feature selection and the machine learning algorithm are independent. The filter feature selection method measures the discriminative ability of each feature, ranks all the features in the original data according to their discriminative ability, and then selects a certain feature according to a predefined threshold to form the final optimal feature subset.

The advantages of this method are that it does not depend on any machine learning method and it is computationally efficient. Therefore, it is suitable for adopting feature selection to large-scale data. While it has the disadvantage that it is separate from the machine learning algorithm, so the characteristics of the machine learning algorithm are not taken into account, therefore, it is difficult to determine whether the selected subset of features can optimize the performance of the classification learning algorithm. The representative Filter approaches are RELIEF [18], FOCUS [19], and MIFS [17]. In our experiments, we use the RELIEF method which will be detailed discussed in the next chapter.

(2) Wrapper approach

The wrapper feature selection method uses a machine learning algorithm to guide the search process for a subset of features and evaluates the generalization and prediction capabilities of the machine learning model for the chosen subset of candidates. Therefore, the wrapper model is also often referred to as a feature selection model based on the search for the best subset of features based on the evaluation results of the machine learning model, so that the subset of features selected by this feature selection method is highly coupled with the machine learning algorithm.

The advantage of this method is that it fits well with the machine learning algorithm since it selects features based on the performance of the machine learning algorithm, thus bringing better results compared with the Filter approach. The disadvantage is that a model needs to be trained for each subset of features to evaluate their merits, so it is computationally intensive, and it is prone to overfitting in case of insufficient samples. For this method, we choose Sequential forward selection (SFS) and Sequential backward selection (SBS) [7], which sequentially select or eliminate the feature set, respectively.

(3) Embedded approach

In the Embedded approach, the process of feature selection and machine learning are

combined together, indicating that the feature selection is also optimized through the process of learning. By doing so, the optimal subset of features is obtained by optimizing the objective function of the machine learning algorithm. In the process of optimizing the machine learning algorithm, the machine learning model removes features that have little impact on the results and keeps the good features in the feature subset.

The method is similar to the Wrapper approach and has the advantage of combining machine learning algorithms with feature selection, and the computational efficiency is high which is similar to the Filter approach. However, the method is susceptible to the effects of the function that optimizes the performance of a subset of features and the settings of its associated parameters, in which the performance and computational efficiency will be significantly influenced [20]. The classical methods in this approach include the LASSO method [21], which adds an L1 penalty term to the regression coefficients to prevent overfitting, specific regression coefficients can be made zero so that a simpler model can be chosen that does not contain those coefficients, and [22] proposed an SVM-RFE based on the support vector machine and recursive feature elimination. This project uses a decision tree-based feature selection method, specifically, random forest as our Embedded approach.

2.2 Dataset

2.2.1 UCI Bank Marketing Data Set [23]

With the wide application of big data technology, banks rely on the intelligent analysis of big data and the accurate judgment of algorithms to carry out diversified and accurate marketing of financial products. Among them, the traditional bank telemarketing method can hardly meet the needs of the times due to the randomness and low hit rate. How to make good use of the various data in the bank database and machine learning technology to improve the accuracy of bank telemarketing is the secret to the success of bank financial products today.

The dataset for this paper was taken from the open-source website UCI and was selected from data related to a marketing campaign conducted by a local Portuguese banking institution [23]. A marketing campaign is the use of telephone calls to one or more telephone contacts to confirm whether a customer will subscribe to a product (bank term deposit) or not. The experimental data consists of 41188 items, including 20 features and 1 label, with the classification objective of predicting whether the customer will subscribe to a time deposit service (variable *y*), corresponding to the classification task. There are 36,548 data items with a "no" label (88.7%) and 4,640 data items with a "yes" label (11.3%).

There are 20 features and 1 label in this experimental dataset, and each feature and its meaning are shown in Table 1 and Table 2. Among the 20 features, half of the variables are categorical and the other half are numerical.

Table 1: Summary of UCI Bank Marketing Data Set

Dataset task	binary classification
Number of features	20
Number of numeric features	10
Number of categorical features	10
Number of observations	41188
Number of normal traffics	36548
Number of attacks	4640

Table 2: Features of UCI Bank Marketing Data Set

No.	feature name	type	definition
1	age	numeric	Age of client
2	job	categorical	Job of client
3	marital	categorical	marital status of client
4	education	categorical	Education status of client
5	default	categorical	Whether the client has default credit
6	housing	categorical	Whether the client has the housing loan
7	loan	categorical	Whether the client has the personal loan
8	contact	categorical	Ways of communication
9	month	categorical	Last contact month
10	day of week	categorical	Last contact day of the week
11	duration	numeric	Last contact time
12	campaign	numeric	contacts during this campaign
13	pdays	numeric	days after last contact
14	previous	numeric	contacts before this campaign
15	poutcome	categorical	outcome of the previous marketing campaign
16	emp.var.rate	numeric	employment variation rate - quarterly indicator
17	cons.price.idx	numeric	consumer price index - monthly indicator
18	cons.conf.idx	numeric	consumer confidence index - monthly indicator
19	euribor3m	numeric	euribor 3 month rate - daily indicator
20	nr.employed	numeric	number of employees - quarterly indicator

2.2.2 Intrusion Detection Evaluation Dataset (CIC-IDS2017) [24]

Network intrusion is one of the greatest threats to the network space and refers to a series of data theft, malicious tampering, and deliberate destruction of computers, networks, programs, and data. In the face of the serious network space security situation, network security situational awareness is increasingly mentioned. Network traffic, as the carrier of information

exchange between endpoints on the Internet, enriches the data flow and controls flow information in the network space. It is of great value for the construction of an intrusion detection system of the network system. The screening of network traffic data anomalies can effectively support the location of intrusions in the network system, especially for the detection of unknown attacks. In the new situation of network space security defense, misuse detection algorithms based on attack signature and pattern matching are increasingly unable to meet the complex security needs in the complex network space, and anomaly detection techniques applied by machine learning algorithms have achieved better results.

As for intrusion detection, the dataset used in this paper is CIC-IDS-2017, which contains both normal traffic and common attacks. The data capture started at 9:00 a.m. on Monday, July 3, 2017, and ended at 5:00 p.m. on Friday, July 7, 2017. This dataset was obtained by the Canadian Institute of Network Security in 2017 by collecting and analyzing simulated network attack traffic and normal traffic. As the dataset is extensive, we only will use 8000 observations corresponding to the day “Thursday, July 6, 2017, Morning” and this report will group all attacks into one category so that our classification task becomes determining whether that network traffic is under network attack, which is a binary classification problem. Here, all normal traffics are labeled as “BENIGH” and all other attacks are labeled as “ATTACK”.

Table 3: Summary of CIC-IDS-2017

Dataset task	binary classification
Number of features	71
Number of numeric features	64
Number of categorical features	7
Number of observations	8000
Number of normal traffics	7930
Number of attacks	70

Table 3 shows the overall characteristics of the CIC-IDS-2017 dataset. As shown in Table 3, one great difference between CIC-IDS-2017 is that the class is highly imbalanced. In detail, there are only 70 observations (0.875%) that are classified as attacks, while 7930 observations (99.125%) are classified as normal traffic. To deal with the class imbalance issue, we had also adopted the oversampling technique SMOTE (Synthetic Minority Oversampling Technique) to add observations for minority classes [25].

2.2.3 Gene Expression Diagnostic (SMK-CAN-187) [26]

With sufficient training samples, deep learning models can learn features for different data

types. feature learning, such as convolutional neural networks for computer vision and BERT for natural language processing. However, deep learning models inherently require a large amount of training data, so it is difficult to achieve good results with small sample data sets. In some application domains, such as medical and security, data collection and labeling are not easy, and the number of valuable data is often in the tens or hundreds, so researchers have started to analyze and study small sample data sets. Gene expression profiling data is characterized as a small sample dataset, and thus deep learning models built on gene expression profiling datasets cannot learn sufficient The predictive power of the models has been dramatically reduced due to the inadequate characterization of gene expression profile data. However, most of the cancer gene expression profiling datasets are HDLSS data, which can also be referred to as microarray data.

For these kinds of data, the dataset we use is SMK-CAN-187, which is a diagnostic gene expression profile. The gene expression data are obtained from smokers with lung cancer and smokers without lung cancer. By analyzing this spectrum using machine learning techniques, we were able to develop a machine learning model for diagnosing smokers with lung cancer, which has substantial clinical benefits. In our experiments, we only selected the first 1000 features as our data to address the limited computational resource.

Table 4: Summary of SMK-CAN-187

Dataset task	binary classification
Number of features	1000
Number of numeric features	1000
Number of categorical features	0
Number of observations	187
Smokers without Lung Cancer	97
Smokers Lung Cancer	90

As shown in Table 4, This type of data is characterized by a large number of features and a small number of samples, and the number of features is even more than the number of samples. In total, The experimental data consists of 187 items, including 1000 features and 1 label, with the classification objective of predicting whether the smoker has lung cancer (variable y), corresponding to the classification task. Here, all features are numeric features corresponding to the specific gene expression.

2.3 Transformer architecture

Recently, attention-based architecture, in particular transformers, which could make use of the correlation of different elements can enhance the element representation, has been widely

adopted in Natural Language Processing [27], Computer Vision [28], Medical imaging analysis [29], and multi-modal fusion [30]. The transformer model is an encoder-decoder architecture as you can see in Figure 3, which includes a number of stacked the transformer encoder module (see left part of Figure 3) and the transformer decoder module (see right part of Figure 3), respectively.

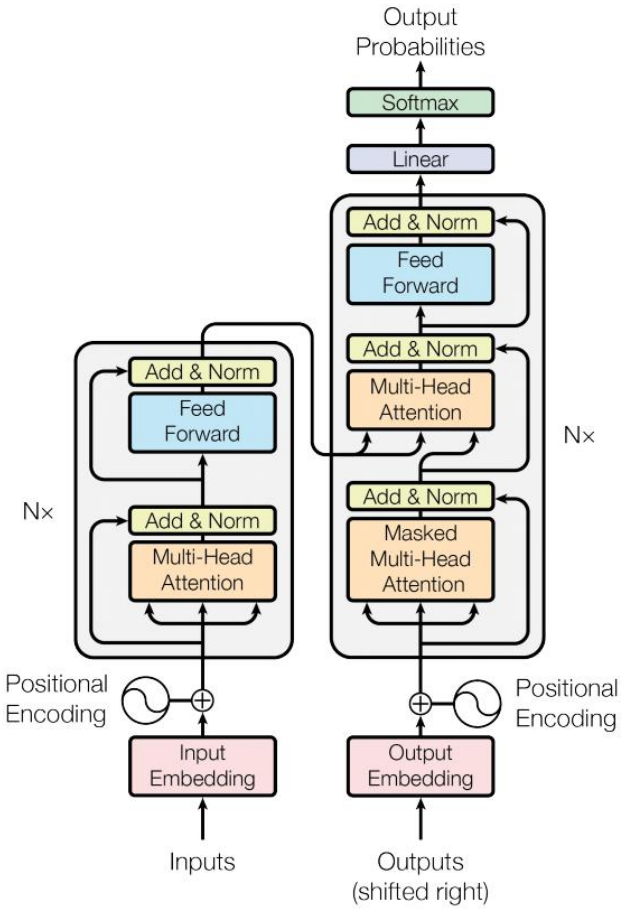


Figure 3: The Transformer - model architecture [27].

In the transformer encoder, the data first passes through a Multi-Head self-attention module as you can see in Figure 4, where multiple heads process input from several different subspaces and are eventually integrated. According to the experiment [27], this design allows the model to learn more informative features by focusing on information from each subspace. self-attention first adopts linear layers to learn the values of Q, K, and V, and then weights each position of the input is learned by Scaled Dot-Product Attention. Specifically, Q, K, V represent Query, Key, and Value respectively, whose concepts are derived from the field of information retrieval. The model matches the corresponding Key in the sequence according to the Query, and finally determines the weight distribution of Value based on the similarity of

the Query and Key. Then, the forward propagation nerve network receives the weight information computed by the Multi-Head self-attention module and processes it by applying the concatenated information to the fully connected layers. To solve the degradation problem in deep learning, the transformer encoder also uses residual connection [31] and layer normalization [32] as shown in Figure 3.

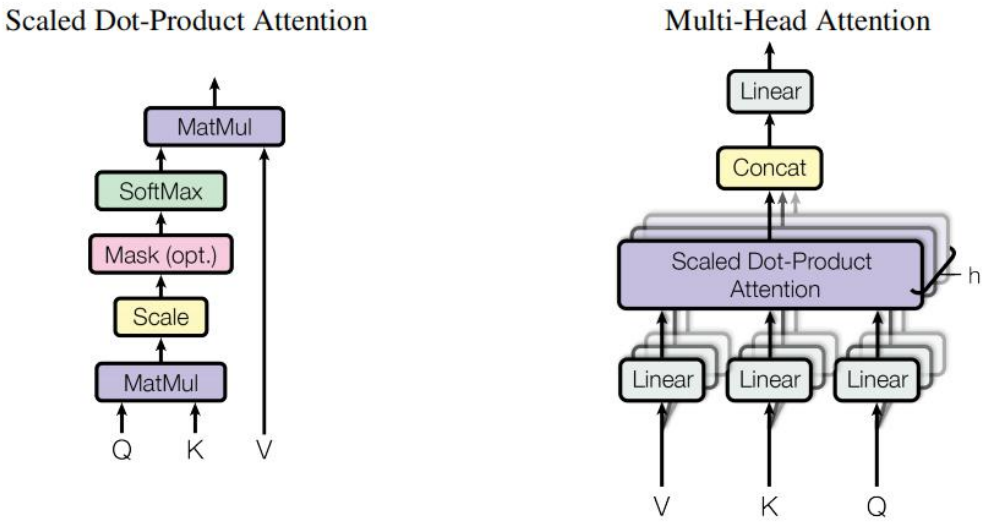


Figure 4: Multi-Head self-attention module [27].

In our project, we only use a transformer encoder to encode raw features in our project, which will be discussed in the implementation of FS-Former.

Chapter 3: Design and Implementation

3.1 Implementation environment

Our project is running on windows 10 platform and implemented by MATLAB R2020b with Statistics and Machine Learning Toolbox and Deep Learning Toolbox. For the implementation of our proposed FSFormer, As for the hardware environment, the CPU is Ryzen 3600, and the GPU is RTX 2070 with 8GB and 16 GB RAM.

3.2 Preprocessing

Data preprocessing is the process of examining, removing, or correcting abnormal data. The purpose of data preprocessing is to change the form of the data to fit and match the needs of the machine learning algorithm. Since there are no missing or duplicate values, therefore, we will only preprocess numerical features and categorical features separately. For CIC-IDS-2017, we also oversampled the data to overcome the issue of class imbalance.

3.2.1 Numeric feature preprocessing

In the field of machine learning, numeric features often have different magnitudes and magnitude units which will make the machine learning model unable to find the optimal stage effectively and accurately. In order to eliminate the influence of magnitudes of numeric features, this paper will dimensionless the data, and the dimensionlessization of data can bring adaptability among numeric features. After adopting dimensionlessization to the original numeric features, they will be in a comparable magnitude, thus training a machine learning algorithm will be much easier, and finally eliminates the influence of dimensionality on the final results. Among them, the most typical are min-max normalization and z-score normalization of the data, the former mapping the original data to between [0, 1] by a linear transformation. However, since our data contains a large number of outliers, the min and max values of the values are very susceptible to the influence of the outliers, and thus can lead to poor results. Therefore, in our project, we adopt Z-score normalization to dimensionlessize our data.

The Z-score, also known as standard deviation normalization, has a mean of 0 and a standard deviation of 1 for the processed data. the transformation formula is:

$$x^* = \frac{x - \overline{x}}{\sigma} \quad (5)$$

where $x*$ is the transformed data, $\overline{x}$ is the mean of raw data and σ is the standard deviation of raw data.

3.2.2 Categorical feature preprocessing

For the categorical feature, there are a finite number of values taken, each representing a category. In addition, for the categorical feature of text type, machine learning algorithms cannot deal with the text directly, and usually, we convert text to numeric values for processing, which requires encoding text as numeric values. In this project, we encode the categorical feature of text type by ordinal encoding. In ordinal encoding, for a feature with *m* categories, we map it correspondingly to the integers [0,*m*-1]. For example, for a feature like "education", we can encode the text value "bachelor", "master", "doctor" as [0,2].

3.2.3 Unbalanced data processing

To address the issue of class imbalance in dataset CIC-IDS-2017, we work on the dataset from the data level by oversampling. Oversampling techniques capture more minority class sample information by increasing the number of minority class samples and improving the underfitting of the dataset.

In this project, SMOTE (Synthetic Minority Oversampling Technique) [33] is used, which oversamples minority class samples. This algorithm is derived from random oversampling techniques which simply increase the minority class samples by copying samples. However, copying samples is very likely to cause the problem of over-fitting. To address the above problem, SMOTE adopts a strategy that artificially synthesizes the minority class new samples by linear interpolation. The steps to generate a new sample are as follows:

(1) For each sample x in the minority class, its k-nearest neighbor is obtained by using Euclidean distance as a criterion to select k samples with minimal distance in the minority class sample set.

(2) For each selected k nearest neighbors, randomly select a set of samples that are represented as x_n.

(3) For each randomly selected sample x_n, a new synthetic sample is generated using random linear interpolation as follows:

$$x_{new} = x + rand(0,1) \times (x_n - x) \quad (6)$$

For this method, we downloaded an implemented package which is publicly available at https://www.ilovematlab.cn/thread-167786-1-1.html.

After adopting SMOTE to dataset CIC-IDS-2017, the class is balanced as shown in Table 5.

Table 5: Summary of CIC-IDS-2017 after oversampling

Dataset task	binary classification
Number of features	71
Number of numeric features	64
Number of categorical features	7
Number of observations	8000
Number of normal traffics	7930
Number of attacks	7930

3.3 Classification

In this paper, we adopted 5 different classification methods, which are Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Random Forest, Support Vector Machine (SVM), and Multi Layer Perceptron (MLP).

3.3.1 Linear Discriminant Analysis (LDA)

LDA classifier is widely used in machine learning as a classical classification method, which is a Gaussian maximum likelihood classification method based on Bayesian decision making. The basic idea of LDA for two-class classification is to find a feature-optimal projection surface to project the features of the training data into a one-dimensional space, and then classify the test samples according to the decision rules. Given a sample $x \in X$, where X is the full dataset, assuming that the mapping function is a linear discriminant function:

$$f(x) = w^T x + w_0 \quad (7)$$

Where x is an h-dimensional feature vector, w is a weight vector, and w_0 is a constant, also known as the threshold weight. w projects the high-dimensional vector x into a one-dimensional space, and w_0 is used to classify the different classes.

The goal of the LDA classifier is to make the projections of similar samples as concentrated as possible and the projections of different classes of samples as dispersed as possible,

assuming that the projection matrix is *W*, and the objective function is:

$$J = \frac{W^T S_b W}{W^T S_w W} \tag{8}$$

where S_b is the between-class scatter matrix and S_w is the within-class scatter matrix, respectively. We also define *C1* as first class and *C2* as second class. Therefore S_b and S_w can be denoted as follows:

$$S_w = S_1 + S_2 = \sum_{x \in C1} (x - \mu_1)(x - \mu_1)^T + \sum_{x \in C2} (x - \mu_2)(x - \mu_2)^T \tag{9}$$

$$S_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \tag{10}$$

where μ_1 and μ_2 are means of samples for *C1* and *C2*, respectively. S_1 and S_2 are covariance matrix for each class.

Let $\|W^T S_w W\| = 1$, the optimal solution *W* is obtained by introducing Lagrange multipliers:

$$S_b W = \lambda S_w W \tag{11}$$

$$W = {S_w}^{-1}(\mu_1 - \mu_2) \tag{12}$$

3.3.2 Quadratic Discriminant Analysis (QDA)

The QDA algorithm is a variant of the LDA algorithm, the difference is that LDA assumes S_1 and S_2 is the same while QDA doesn't. Due to the above difference, LDA separates the data with a linear surface, while QDA separates the data with a quadratic surface.

3.3.3 Support Vector Machine (SVM)

Support Vector Machines (SVM) were first proposed by Vapnik in 1995 [33], and have been developed and explored for many years. SVM is now used in a variety of fields such as pattern recognition and nonlinear regression. The purpose of support vector machines is to determine a hyperplane to classify a data set, and the closest data to the hyperplane in each class are the "support vectors". The method of determining the hyperplane is to maximize the

sum of the distances from these "support vectors" to the hyperplane using optimization methods. In our project, we adopt linear separable support vector machines.

For a dataset $X = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}, y_i \in \{-1,1\}$, where X is the input of the SVM model and y_i is the classification outcomes. SVM is based on finding a hyperplane in the training set sample space that completely separates the samples with different classification results. This target hyperplane can be represented by the following linear equation:

$$w^T x + b = 0 \tag{13}$$

where w is the normal vector and b is the displacement, these two parameters determine the direction and intercept of the hyperplane, thus determining the position of the plane. Let the hyperplane be (w, b), any point in the sample space x_i to the hyperplane can be expressed as:

$$r = \frac{|w^T x + b|}{\|w\|} \tag{14}$$

If the hyperplane yields no sample misclassification, then all samples in the training set can satisfy the following: If $y_i = 1,\ w^T x + b > 0$, and If $y_i = -1,\ w^T x + b < 0$. Let:

$$\begin{cases} w^T x + b \geq +1, y_i = +1; \\ w^T x + b \leq -1, y_i = -1; \end{cases} \tag{15}$$

The samples such that the equal relation holds are the sample points with the smallest distance from the target hyperplane, and these sample points are the "support vectors", and the margin is:

$$\gamma = \frac{2}{\|w\|} \tag{16}$$

The target hyperplane is the plane that achieves the maximum margin, which is the plane that minimizes $\frac{1}{\gamma}$:

$$\min_{w,b} \frac{\|w\|^2}{2}$$
$$s.t.\ y_i(w^T x + b) \geq +1 \tag{17}$$

In order to solve Eq. (17), the Lagrange multiplier method is usually used to obtain the "dual problem" of this equation, Lagrangian function is:

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} + \sum_{i=1}^{m} \alpha_i (1 - y_i(w^T x_i + b)) \tag{18}$$

And its dual problem is:

$$\max_{w,b} \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j\, y_i y_j x_i{}^T x_j$$
$$s.t. \sum_{i=1}^{m} \alpha_i\, y_i\ = 0, \tag{19}$$
$$\alpha \geq 0$$

After solving α, then we can obtain the parameter of the SVM model:

$$f(x) = w^T x + b = \sum_{i=0}^{m} \alpha_i\, y_i x_i{}^T X + b \tag{20}$$

3.3.4 Random Forest (RF)

The decision tree-based random forest algorithm proposed by Leo Breiman [14] is one of the widely used integrated learning algorithms today. Its core idea is to combine Bagging [35] integrated learning theory with the random subspace method [36], which has higher classification accuracy compared with the traditional decision classification tree algorithm.

Random forest is based on Bagging integration theory, which uses decision classification trees as sub-classifiers. Firstly, the Bootstrap random sampling technique is used to generate

multiple sub-training sets and their test sets from the dataset with put-back sampling, and then, independent decision trees are constructed for each sub-training set to construct a random forest.

To construct a random forest, suppose there are m samples with n features, the number of decision trees in the random forest is k. Briefly, the process is as follows:

(1) The bootstrap method is used to sample m samples from the dataset with put-back to set up k sub-training datasets. The rest of the unsampled samples are used as k out-of-bag (OOB) data.

(2) The classification tree is used to construct the sub-classifier. For each node in the decision tree, a correlation criterion is adopted to select m segmentation features randomly. After that, the chosen node will be divided into 2 sub-nodes with the optimal segmentation features and optimal segmentation points. This segmentation will be cycled until there are no nodes that could be segmented, that is, all nodes are leaf nodes.

(3) Repeat k times the above steps to form k decision trees, which will be assembled into a random forest.

After the random forest is constructed, it can be used for prediction with the following process:

(1) Classify the dataset X using k decision trees in the random forest to obtain k predictions.

(2) The plural of each decision tree prediction result is used as the final prediction result.

3.3.5 Multi Layer Perceptron (MLP)

MLP, also known as deep forward network, is a typical deep learning method, and the original deep learning is using neural networks to extract features. the purpose of MLP is to approximate a function f that maps an input x to a response y and learns the value of the parameter theta so that it can get the best approximation of the function. A single perceptron in MLP is shown in Figure 5.

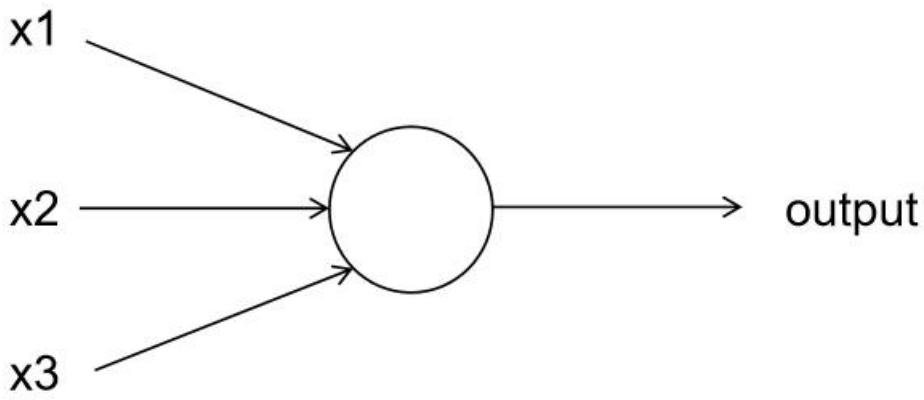


Figure 5: Perceptron

Equation of operation in perceptron is:

$$output = \sum_i w^i x^i \tag{21}$$

MLP generally consists of three layers: input layer, hidden layer, and output layer. As shown in Figure 2, from left to right are the input layer, hidden layer, and output layer.

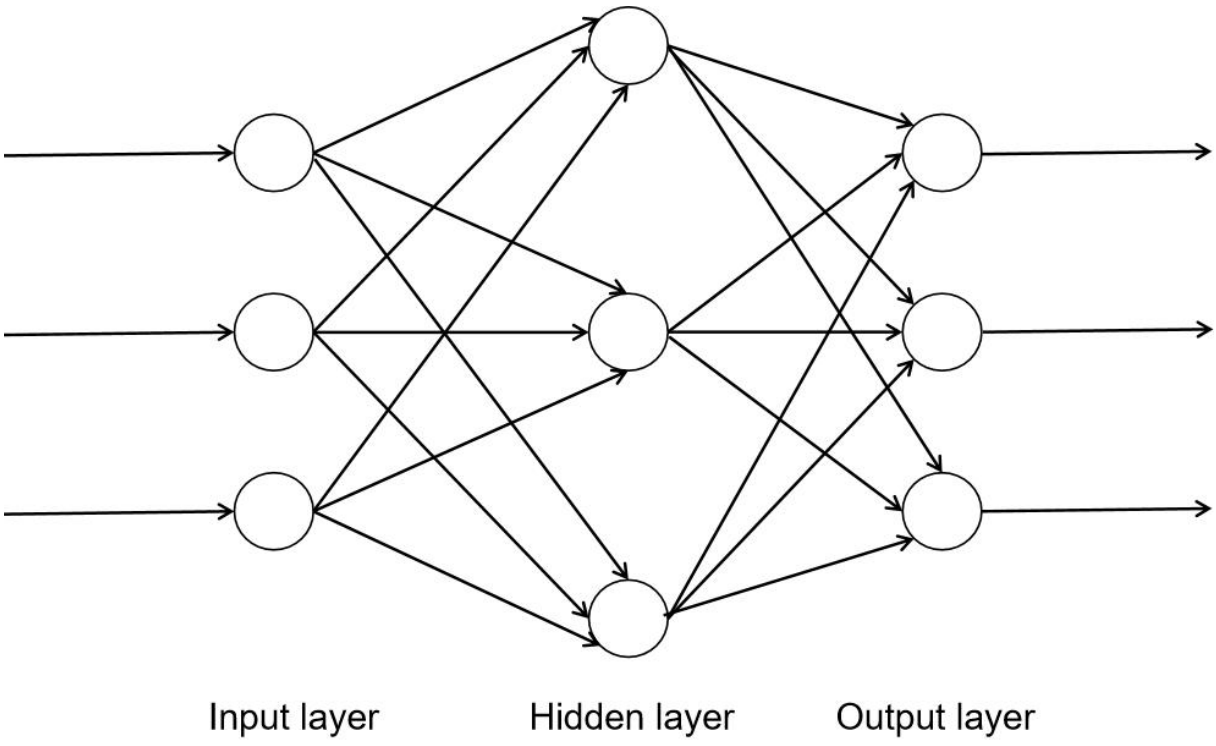


Figure 6: Illustration of MLP

To update and optimize the MLP and other deep learning methods, a typical approach is Gradient descent which uses backpropagation strategy to update the parameters of the networks.

In this project, our MLP is structured as shown in Figure 7.

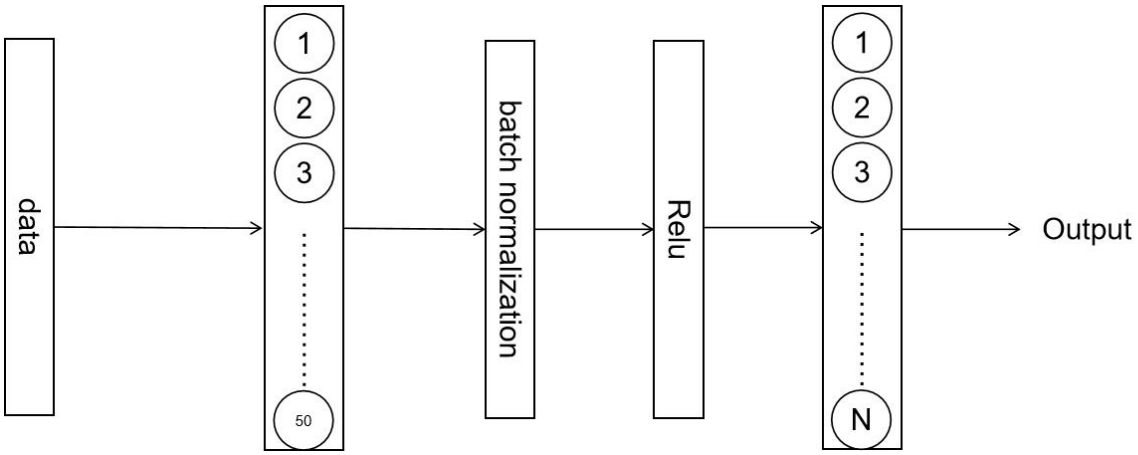


Figure 7: Structure of MLP in this project.

In order to scale the features in the MLP, we adopt batch normalization in the middle, which is mathematically represented as:

$$X = \frac{X - \mu}{\sigma} \tag{22}$$

where X is a batch of samples and μ and σ are the means and variance of X.

We also introduce the activation function Relu, which is the key to allowing the neural network to fit a nonlinear function. Without the activation function, no matter how many layers there are in the hidden layer, it can only be fitted to a linear function in the end. Relu can be mathematically represented as:

$$x = \begin{cases} x, x > 0; \\ 0, x \leq 0; \end{cases} \tag{23}$$

3.4 Feature selection

3.4.1 Filter approaches

In the filter feature selection method, the feature selection and the classification algorithm are two independent processes. The features in the dataset are first filtered according to certain criteria, and then, the filtered features are used to train the classifier. In this project, we implement the Relief method to select features.

The relief method is one of the most commonly used feature selection methods. The method determines the goodness of features based on their relevance to the label and then removes the unsuitable features. The first proposed Relief algorithm focuses on the binary classification problem, which adopts a "correlation statistic" to measure the importance of features, which is a vector, and each element of the vector is the evaluation value of one of the initial features.

The importance is a relevant measure for each feature in the subset, so it can be seen that this "relevant statistic" can also be considered as the "weight" of each feature. You can specify a threshold τ and select only the feature value corresponding to the correlation statistic larger than τ. You can also choose the number of features you want to select denoted as *k*, and then select the *k* features with the largest importance measures.

In the Relief method, for each $x_i \in X$, where X is the whole dataset and x_i is *ith* sample, Define the nearest sample of the same class as *nearHit*, and the nearest sample of different class as *nearMiss*. Then the algorithm is:

(1) Randomly select a sample $x_i \in X$.

(2) For the given x_i, find *k nearHit* and *nearMiss.*

(3) Update the weights with the following equation:

$$W_I = W_{I-1} - \frac{\sum_{j=1}^{k} diff(x_i^j, nearHit_j)^2}{k} + \frac{\sum_{j=1}^{k} diff(x_i^j, nearMiss_j)^2}{k} \quad (24)$$

where $diff()$ represents the distance between x_i and $nearHit_i$ or $nearMiss_i$. Which is defined as:

(1) For categorical features:

$$diff(x, y) = \begin{cases} 0, x = y \\ 1, otherwise \end{cases} \quad (25)$$

(2) For numeric features:

$$diff(x, y) = |x - y| \quad (26)$$

In this project, we set the number of *k* to be 10, and we select features with weights>τ.

3.4.2 Wrapper approaches

In Wrapper approaches, the optimal feature subset is selected based on the evaluation performance of the implemented classifier. For this method, we choose Sequential forward selection (SFS) and Sequential backward selection (SBS), which sequentially select or eliminate the feature set, respectively.

3.4.2.1 Sequential forward selection (SFS)

The SFS feature selection starts with the empty set and selects one feature at a time, and then feature f_i is added to the feature set F such that the objective function is optimal. The process of SFS is as follows:

(1) Determine the empty set of features F_0.

(2) Add a feature f_i to the current feature set F_k, in which f_i satisfies:

$$f_i = \underset{f_i}{argmax}\, obj(F_k + f_i) \tag{27}$$

where $F_k + f_i$ represents adding the *ith* feature to the current feature set F_k, and k is the number of iterations. $obj()$ is the objective function.

(3) Update current feature set F_k:

$$F_{k+1} = F_k + f_i \tag{28}$$

3.4.2.2 Sequential backward selection (SBS)

The SBS feature selection is the opposite of the SFS algorithm, which starts from the full set of features and then continuously discards features from the feature set to achieve the optimal value of the objective function. The process of SBS is as follows:

(1) Determine the full set of features F_0.

(2) Delete a feature f_i from the current feature set F_k, in which f_i satisfies:

$$f_i = \underset{f_i}{argmax}\, obj(F_k - f_i) \tag{29}$$

where $F_k - f_i$ represents adding *i*th feature to the current feature set F_k, and k is the number of iterations. $obj()$ is the objective function.

(3) Update current feature set F_k:

$$F_{k+1} = F_k - f_i \tag{30}$$

3.4.3 Embedded approaches

For the embedded feature selection approaches, this project has adopted a random forest. To select the most informative features, the importance of a feature *f* in the random forest should be calculated as follows:

(1) It first calculates the OOB (OOB, the data which is not sampled in the construction process) data error of every decision tree in the random forest, denoted as $OOBerr_1$.

(2) Adding random noise to the samples with feature *f* of all OOB data. Then calculate the corresponding noised data error, denoted as $OOBerr_2$.

(3) Assuming that there are *K* trees in the random forest, the feature importance is:

$$\text{Importance(X)} = \sum_{i}^{k} (OOBerr_1^i - OOBerr_2^i)/K \qquad (31)$$

The intuition to adopt this equation to measure the importance of a feature is that: If a feature is disturbed with random noise, the corresponding out-of-bag accuracy will be greatly reduced, indicating that this feature has significantly impacted the classification results, therefore, the importance of this feature will be measured by the degree of performance damage. Then we select features with $Importance(X) > 0$.

3.4.4 FS-Former

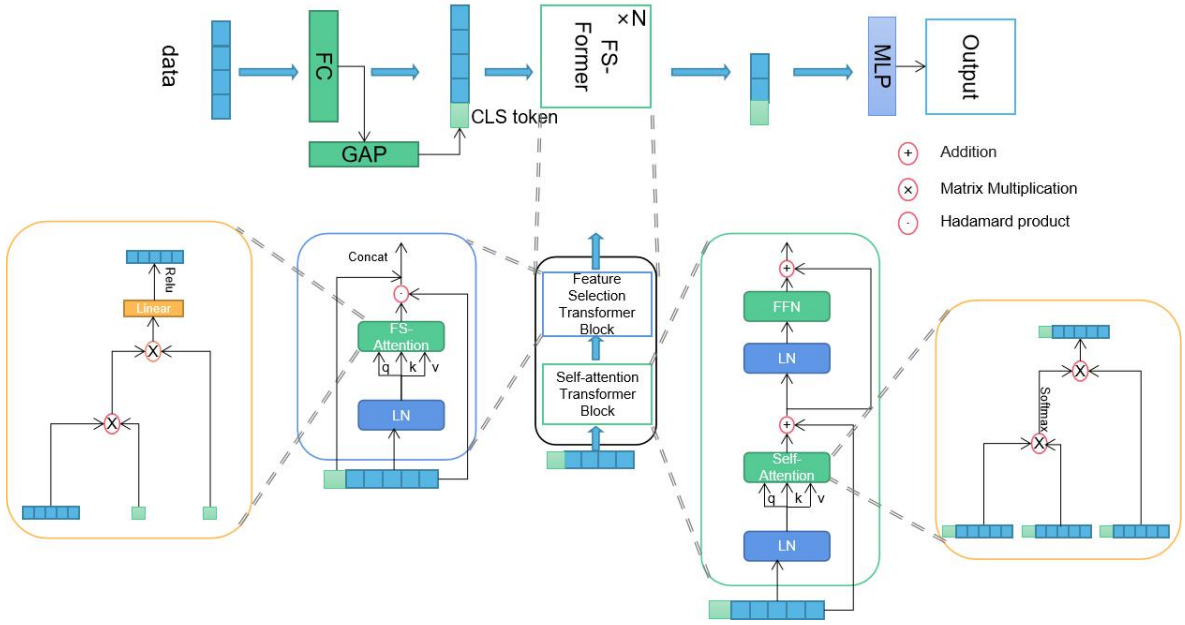


Figure 8: Overview of FS-Former framework.

In this project, we also propose a wrapper feature selection method based on transformer architecture as shown in Figure 8. In the FS-Former framework, the data first passes through an FC layer to be embedded. Then we use the global average pooling operation in which all values of features are summed and averaged to obtain a value. Here this value is the "CLS token", which is the abstract global information of all features. Then we concatenate the CLS token with other feature vectors.

Then comes the core part of the proposed method, the feature selection transformer short for FS-Former. which contains a Feature Selection Transformer Block and a Self-attention Transformer Block. Here the Self-attention Transformer Block is borrowed from the classical transformer encoder architecture, which is well known in the area of deep learning.

As for Feature Selection Transformer Block, all features are used as q, to query k which is derived from the abstract global information CLS token. After that, an attention map for each feature is obtained. Then the attention map is used to map the v which is also derived from the CLS token from the perspectives of each feature and then linear project the outcome to get a weight for each feature. Afterward, the Relu function will gate the weights. Eventually, the gated weights will Hadamard product with the original features. By doing so, irrelevant features are eliminated since the corresponding attention weights are set to 0. Therefore, only

the most informative features are selected. This process can be mathematically represented as:

$$Q = L_Q X, K = L_K X_{cls}, V = L_V X_{cls},$$

$$A = \frac{QK^T}{\sqrt{C}} \tag{32}$$

$$W = Relu(L_P(AV))$$

$$X = W \times X$$

Where L_Q, L_K and L_V are learnable linear functions that project input in the same dimensions. And C is the dimension of input. And L_P is another learnable linear function that projects input to 1 dimension.

Finally, the selected features and CLS token will feed into MLP as explained in Section 3.3.5.

3.5 Tuning and debugging of the methods

There are 3 methods that need to be tuned which are RF and Relief and our proposed FS-Former.

3.5.1 RF tuning

We can tune the parameter NumLearningCycles, which is the classification trees included in the random forest. Here we select this parameter from (1,100) to different datasets. The tuning figures for each dataset are shown in Figure 9:

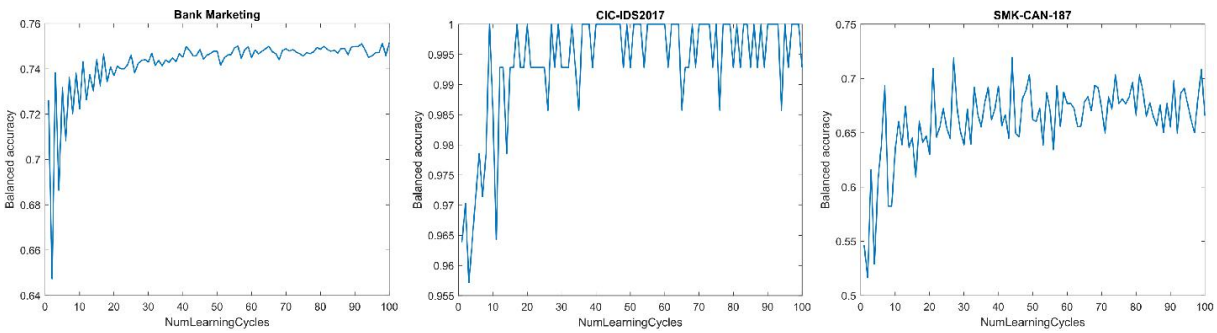


Figure 9: NumLearningCycles tuning.

After tuning, the NumLearningCycles we select are shown in Table 6.

Table 6: Optimal NumLearningCycles in different dataset.

dataset	NumLearningCycles
Bank Marketing	55
CIC-IDS2017	9
SMK-CAN-187	44

3.5.2 Relief tuning

We also tuned the Relief by searching for the best parameter "Number of nearest neighbors", the searching space is from 10 to 100, with a step 10. The tuning figure for each dataset is shown in Figure 10:

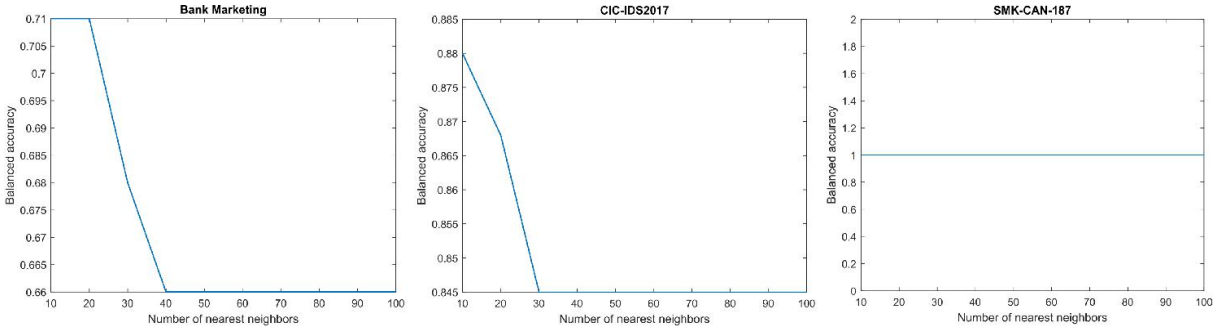


Figure 10: Number of nearest neighbors tuning.

After tuning, the Number of nearest neighbors we select is shown in Table 7.

Table 7: Optimal Number of nearest neighbors in different datasets.

dataset	Number of nearest neighbors
Bank Marketing	10
CIC-IDS2017	10
SMK-CAN-187	10

3.5.3 FS-Former

To explore the optimal hyper-parameter settings in FS-Former, we adjusted learning rate (lr) and weight decay through a grid search strategy. These parameters that had tried in Bank Marketing dataset as shown in

Table 8: Hyper parameters settings of FS-Former.

Hyper parameters settings	Balanced accuracy
lr=0.01; weight_decay=$5e^{-9}$	0.86924
lr=0.01; weight_decay=$5e^{-4}$	0.75383
lr=0.001; weight_decay=$5e^{-9}$	0.74221
lr=0.001; weight_decay=$5e^{-4}$	0.78266
lr=0.0001; weight_decay=$5e^{-9}$	0.7277
lr=0.0001; weight_decay=$5e^{-4}$	0.71883

Therefore, we choose *lr*=0.01 and weight_decay=$5e^{-9}$ as our hyper-parameter setting of FS-Former.

3.6 5-fold Cross-Validation

In order to estimate the variability of the results, Montecarlo experiments have been implemented through 5-fold Cross-Validation. Thus the mean and standard deviation of the results of the experiments can be estimated and discussed.

The process of 5-fold Cross Validation is:

(1) The full dataset is randomly separated into 5 copies without being sampled repeatedly.

(2) For each fold, the selected copy is adopted as the test set, and the rest 4 copies are used as the training set.

(3) Repeat 5 times, so that each copy will be used as the test set and the others as the training set. Therefore this project will get a trained model for each fold.

(4) For each fold, the corresponding evaluation metrics on the test set are obtained and the means and standard deviation corresponding metrics are calculated on 5 folds as an estimate.

Chapter 4: Results and Discussion

In this chapter, we have evaluated and compared the different methods in terms of performance, computational cost as well as features they selected on different datasets. Specifically, the datasets we evaluated are Bank Marketing, CIC-IDS2017, CIC-IDS2017 with oversampling, and SMK-CAN-187.

4.1 Performance Evaluation

Here, we evaluated the performance of the different implemented methods. The evaluation contains four metrics that are accuracy (ACC), balanced accuracy (BACC), the area under curve (AUC), sensitivity (SENS), and specificity (SPEC), which can be mathematically described as follows:

$$\begin{aligned} \mathrm{ACC} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \\ \mathrm{SENS} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{SPEC} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \\ \mathrm{BACC} &= \frac{\mathrm{SENS}+\mathrm{SPEC}}{2} \\ \mathrm{AUC} &= \text{Area under receiver operating characteristic curve} \end{aligned} \tag{33}$$

Here, We adopted a Cross-Validation strategy to evaluate the performance. As shown in Tables 9, 10, 11, and 12, which contain the mean values of all splits, the standard deviation is in the corresponding brackets.

Table 9: Performance of different methods on Bank Marketing.

Feature selection	Classification	BACC	ACC	AUC	SENS	SPEC
Full data (using all features)	LDA	0.7239(0)	0.9088(0)	0.7239(0)	0.4851(0)	0.9626(0)
	QDA	0.7579(0)	0.8746(0)	0.7579(0)	0.6073(0.014)	0.9086(0)
	RF	0.7449(0)	0.9142(0)	0.7449(0)	0.5265(0.01)	0.9634(0)
	SVM	0.6378(0)	0.9024(0)	0.6378(0)	0.2963(0.01)	0.9793(0)
	MLP	0.7153(0.01)	0.9116(0)	0.7153(0.01)	0.4619(0.02)	0.9687(0)
SFS(Embedded)	LDA	0.7310(0.01)	0.9096(0)	0.7310(0.01)	0.5004(0.022)	0.9615(0)
	QDA	0.7847(0)	0.8834(0)	0.7847(0)	0.6573(0)	0.9121(0)
	RF	0.7560(0.01)	**0.9150(0)**	0.7560(0.01)	0.5506(0.02)	0.9613(0)
	SVM	0.5967(0)	0.8975(0)	0.5967(0)	0.2084(0)	0.985(0)
	MLP	0.5(0)	0.8873(0)	0.5(0)	0(0)	1(0)
SBS(Embedded)	LDA	0.7426(0.01)	0.9084(0)	0.7426(0.01)	0.5287(0.02)	0.9566(0)
	QDA	0.7846(0)	0.8715(0)	0.7846(0)	0.6724(0)	0.8968(0)
	RF	0.7475(0.014)	0.9140(0)	0.7475(0.014)	0.5325(0.026)	0.9624(0)
	SVM	0.7045(0.022)	0.9107(0)	0.7045(0.022)	0.4384(0.046)	0.9707(0)
	MLP	0.6662(0)	0.9090(0)	0.6662(0)	0.3528(0)	0.9796(0)
Relief	LDA	0.6835(0.01)	0.9032(0)	0.6835(0.01)	0.4000(0.022)	0.9671(0)
	QDA	0.6968(0.01)	0.8957(0)	0.6968(0.01)	0.4401(0.022)	0.9535(0)
	RF	0.7385(0.01)	0.9103(0)	0.7385(0.01)	0.5168(0.02)	0.9603(0)
	SVM	0.5913(0.01)	0.8967(0)	0.5913(0.01)	0.1972(0.014)	**0.9855(0)**
	MLP	0.6646(0.017)	0.9042(0)	0.6646(0.017)	0.3554(0.036)	0.9739(0)
RF		0.7491(0)	0.9149(0)	0.7491(0)	0.5351(0)	0.9631(0)
FS-Former		**0.8692(0)**	0.8428(0.01)	**0.9114(0.01)**	**0.9007(0.017)**	0.8378(0.017)

Table 10: Performance of different methods on CIC-IDS2017.

Feature selection	Classification	BACC	ACC	AUC	SENS	SPEC
Full data (using all features)	LDA	0.7839(0.0262)	0.9786 (0.01)	0.7839(0.0262)	0.9821(0.01)	0.5857(0.1107)
	QDA	0.5000(0)	0.9913(0)	0.5000(0)	**1(0)**	0(0)
	RF	0.9929(0.017)	0.9999(0)	0.9929(0.017)	**1(0)**	0.9857(0)
	SVM	0.9998(0)	0.9996(0)	0.9998(0)	0.9996(0)	**1(0)**
	MLP	0.6141(0.0252)	0.9929(0)	0.6141(0.0252)	0.9996(0)	0.2286(0.1005)
SFS(Embedded)	LDA	0.8341(0.071)	0.9799(0)	0.8341(0.071)	0.9825(0)	0.6857(0.0194)
	QDA	0.9000(0.0500)	0.9982(0)	0.9000(0.0500)	1(0)	0.8000(0.2000)
	RF	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	SVM	0.5000(0)	0.9913(0)	0.5000(0)	**1(0)**	0(0)
	MLP	0.5000(0)	0.9913(0)	0.5000(0)	**1(0)**	0(0)
SBS(Embedded)	LDA	0.8453(0.2)	0.9740(0)	0.8453(0.2)	0.9763(0)	0.7143(0.0153)
	QDA	0.5890(0)	0.1852(0)	0.5890(0)	0.1780(0)	1(0)
	RF	0.9642(0.5)	0.9993(0)	0.9642(0.5)	0.9999(0)	0.928(0.0102)
	SVM	0.9999(0)	0.9998(0)	0.9999(0)	0.9997(0)	1(0)
	MLP	0.7499(0.0172)	0.9954(0)	0.7499(0.0172)	0.9997(0)	0.5000(0.0689)
Relief	LDA	0.8407(0.052)	0.9790(0)	0.8407(0.052)	0.9815(0.022)	0.7000(0.0112)
	QDA	0.5000(0)	0.9913(0)	0.5000(0)	**1(0)**	0(0)
	RF	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	SVM	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	MLP	0.6143(0.0251)	0.9933(0)	0.6143(0.0251)	**1(0)**	0.2286(0.1005)
RF		0.9857(0.031)	0.9998(0)	0.9857(0.031)	**1(0)**	0.9714(0.064)
FS-Former		0.9961(0)	0.9961(0)	0.9963(0)	0.9926(0)	0.9997(0)

Table 11: Performance of different methods on CIC-IDS2017 with SMOTE.

Feature selection	Classification	BACC	ACC	AUC	SENS	SPEC
Full data (using all features)	LDA	0.9697(0)	0.9697(0)	0.9697(0)	0.9396(0)	0.9999(0)
	QDA	0.9998(0)	0.9998(0)	0.9998(0)	**1(0)**	0.9996(0)
	RF	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	SVM	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	MLP	0.9996(0)	0.9996(0)	0.9996(0)	0.9992(0)	**1(0)**
SFS(Embedded)	LDA	0.9714(0.022)	0.9714(0.022)	0.9714(0.022)	0.9428(0.043)	**1(0)**
	QDA	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	RF	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	SVM	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	MLP	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
SBS(Embedded)	LDA	0.9770(0)	0.9770(0)	0.9770(0)	0.9542(0)	0.9997(0)
	QDA	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	RF	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	SVM	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	MLP	0.9996(0)	0.9996(0)	0.9996(0)	0.9991(0)	**1(0)**
Relief	LDA	0.9532(0)	0.9532(0)	0.9532(0)	0.9087(0)	0.9976(0)
	QDA	0.9961(0)	0.9961(0)	0.9961(0)	1(0)	0.9922(0)
	RF	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	SVM	**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
	MLP	0.9999(0)	0.9999(0)	0.9999(0)	0.9997(0)	**1(0)**
RF		**1(0)**	**1(0)**	**1(0)**	**1(0)**	**1(0)**
FS-Former		0.9913(0.014)	0.9913(0.014)	0.9950(0)	0.9851(0.026)	0.9975(0)

Table 12: Performance of different methods on SMK-CAN-187.

Feature selection	Classification	BACC	ACC	AUC	SENS	SPEC
Full data (using all features)	LDA	0.6511(0.064)	0.6531(0.2)	0.6511(0.064)	0.7021(0.0120)	0.6000(0.0145)
	QDA	0.6174(0.0121)	0.6141(0.0122)	0.6174(0.0121)	0.5237(0.0259)	0.7111(0.072)
	RF	0.6572(0.064)	0.6580(0.064)	0.6572(0.064)	0.6811(0.072)	0.6333(0.2)
	SVM	0.7149(0.081)	0.7171(0.0061)	0.7149(0.081)	0.7632(0.067)	0.6667(0.0216)
	MLP	0.6625(0.034)	0.6632(0.034)	0.6625(0.034)	0.6805(0.066)	0.6444(0.03)
SFS(Embedded)	LDA	0.7427(0.049)	0.7437(0.045)	0.7427(0.049)	0.7521(0.0129)	0.7333(0.0238)
	QDA	**0.7747(0.062)**	**0.7751(0.2)**	**0.7747(0.062)**	0.7716(0.0124)	**0.7778(0.078)**
	RF	0.7585(0.043)	0.7592(0.043)	0.7585(0.043)	0.7837(0.053)	0.7333(0.072)
	SVM	0.7477(0.06)	0.7489(0.0036)	0.7477(0.06)	0.7732(0.057)	0.7222(0.096)
	MLP	0.6833(0.037)	0.6846(0.037)	0.6833(0.037)	0.7111(0.047)	0.6556(0.072)
SBS(Embedded)	LDA	0.7516(0.073)	0.7538(0.074)	0.7516(0.073)	**0.7811(0.0370)**	0.7222(0.0231)
	QDA	0.6517(0.066)	0.6521(0.067)	0.6517(0.066)	0.6479(0.0140)	0.6556(0.0114)
	RF	0.6814(0.066)	0.6844(0.067)	0.6814(0.066)	0.7405(0.0185)	0.6222(0.0114)
	SVM	0.7146(0.028)	0.7168(0.028)	0.7146(0.028)	0.7737(0.067)	0.6556(0.046)
	MLP	0.6602(0.071)	0.6637(0.067)	0.6602(0.071)	0.7426(0.0038)	0.5778(0.0194)
Relief	LDA	0.6416(0.065)	0.6420(0.067)	0.6416(0.065)	0.6611(0.0210)	0.6222(0.082)
	QDA	0.6989(0.031)	0.7003(0.031)	0.6989(0.031)	0.7311(0.072)	0.6667(0.055)
	RF	0.6977(0.0128)	0.7017(0.0116)	0.6977(0.0128)	0.7842(0.088)	0.6111(0.0509)
	SVM	0.7435(0.037)	0.7435(0.037)	0.7435(0.037)	0.7426(0.095)	0.7444(0.074)
	MLP	0.7005(0.046)	0.7010(0.047)	0.7005(0.046)	0.7121(0.072)	0.6889(0.2)
RF		0.6813(0.057)	0.6848(0.055)	0.6813(0.057)	0.7626(0.03)	0.6000(0.099)
FS-Former		0.7090(0.022)	0.7111(0.0006)	0.7152(0.014)	0.7187(0.0148)	0.6993(0.0084)

As you can see in the above Tables 9, 10, 11, and 12, the Filter approach Relief has lower performance compared with the Wrapper approach SFS or SBS in most cases. And Embedded approach RF shows the worst performance compared with other feature selection methods. But the performance of all of the feature selection methods is greater than just using the classier without feature selection. This shows that our implemented feature selection procedure successfully selects the most informative features then improve the corresponding

classifier's performance. We also evaluate the effectiveness of data oversampling and necessity of balanced data for feature selection as shown in Table 10 and 11. The corresponding results demonstrate that the performance many methods which sensitive to the data balance significantly increase from SMOTE oversampling techniques, indicating the importance of adopting oversampling techniques to imbalanced data for feature selection.

For our proposed FS-Former, we can see that compared with MLP with full data or other feature selection methods, our methods achieve comparable performance, suggesting the effectiveness of our FS-Former to filter the redundant features. Specifically, in Bank Marketing, FS-Former even surpasses all other methods. However, from the results, we can see that there is no specific method superior to all other methods and each method has its own characteristics and advantages, therefore, we can not easily conclude which method is the best.

We also draw the confusion matrix and ROC curve for different cases, As ROC curves show the performance of the method for different levels of probability of false al arm (PFA). One method could perform better in some ranges of PFA, but perform worse in ot her ranges of PFA. However, since there are extensive figures, we only show LDA on Bank Marketing in this report (see Figure 11 and Figure 12) while the full figures will be shown in the supplementary material.

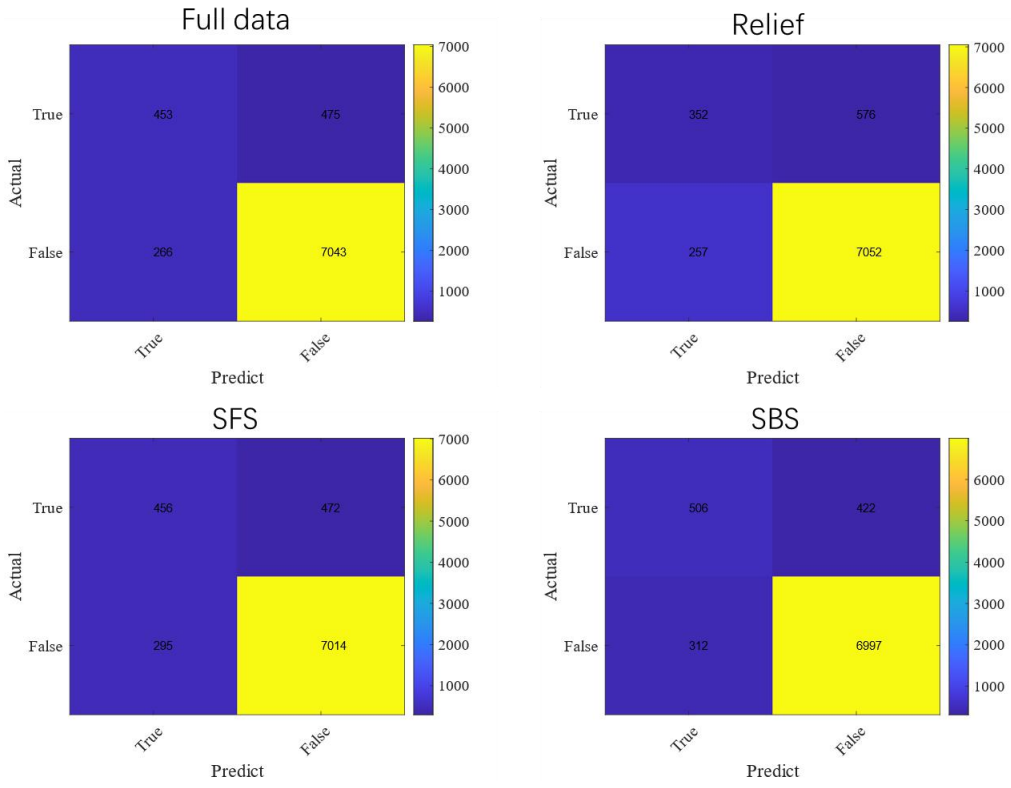


Figure 11: Confusion matrix of LDA on Bank Marketing.

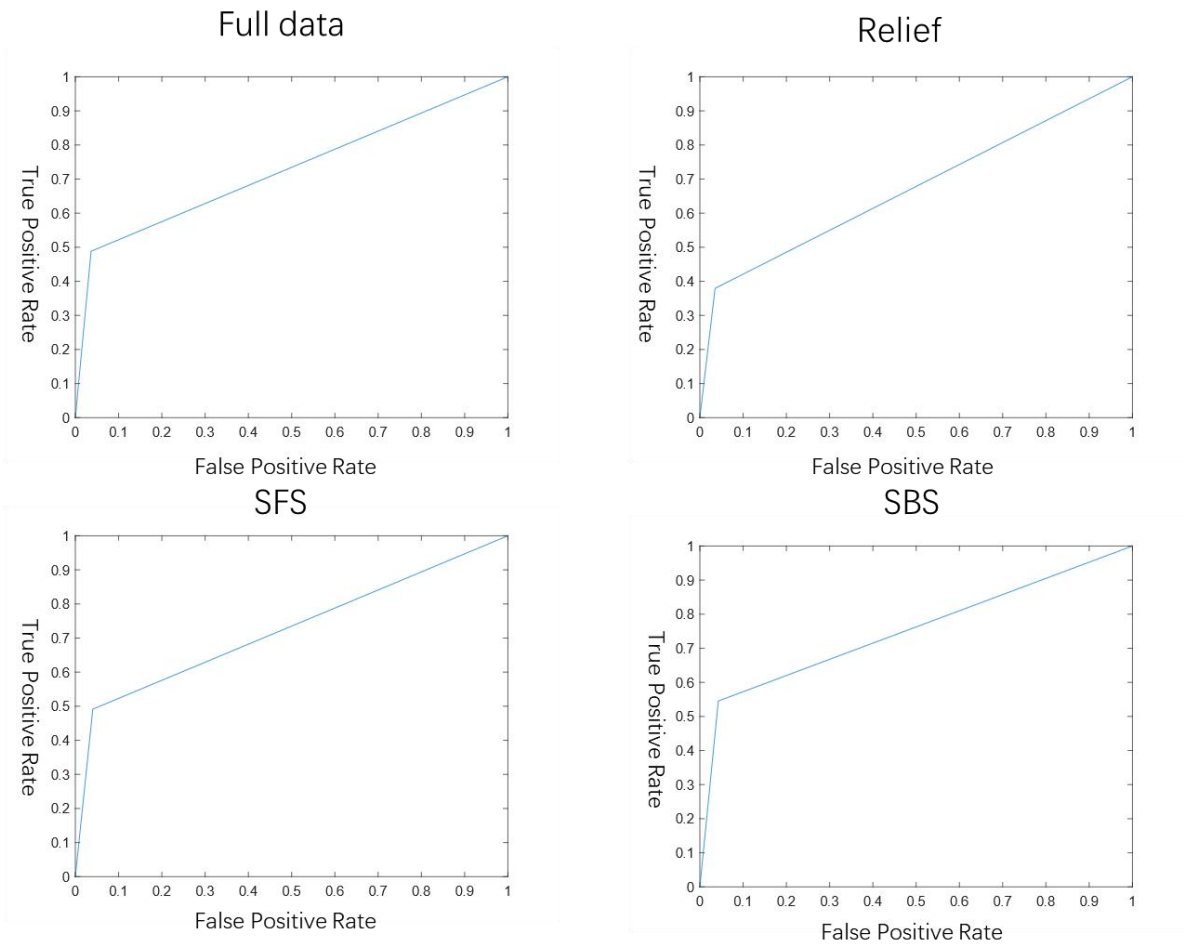


Figure 12: ROC curve of LDA on Bank Marketing.

As shown in Figure 11, the confusion matrix can successfully reveal the behavior of the classifier by indicating the classification results of each class. In Figure 12, it is clear to see that the ROC curve can fully exploit the overall performance of the classifier by changing the threshold, and thus find an optimal classification threshold.

4.2 Computational cost Evaluation

In the experiments, we also evaluate the computational cost in terms of running time in 3 aspects: training time, inference time, and feature selection time.

As shown in Table 13 and Table 14, we find that in most cases, feature selection methods can not only improve the classification performance but also improve the computational efficiency marginally. The reason why the feature selection could reduce the computational cost is that the computation time for machine learning algorithm is proportional to the number of features. As for the proposed FS-Former, since the computational complexity for transformer architecture is high, the computational cost is expensive, suggesting we should optimize the architecture of the proposed FS-Former in future work. Besides, by comparing

MLP and FS-Former with other classical machine learning classification algorithms, the results show that the computational training cost for deep learning methods is expensive, therefore, it is necessary to adopt feature selection to deep learning methods to reduce that costs.

Table 13: Training time of different methods.

Feature selection	Classification	Bank Marketing	CIC-IDS2017	CIC-IDS2017 with SMOTE	SMK-CAN-187
Full data (using all features)	LDA	0.204607s	0.196794s	0.555903s	0.26922s
	QDA	0.428145s	0.200563s	0.392538s	0.175821s
	RF	26.777703s	0.549203s	0.932403s	1.3638s
	SVM	1.107046s	0.549599s	1.396226s	0.159162s
	MLP	120.24567s	44.4326s	73.2749s	27.7102s
SFS(Embedded)	LDA	**0.101998s**	**0.042585s**	0.12969s	0.284061s
	QDA	0.244226s	0.106122s	0.114275s	**0.049882s**
	RF	21.627891s	0.35255s	0.476127s	1.022013s
	SVM	0.261778s	0.182769s	0.183438s	0.16697s
	MLP	118.9993s	46.2536s	66.8277s	22.8048s
SBS(Embedded)	LDA	0.390063s	0.640714s	0.285203s	0.071533s
	QDA	0.169761s	0.143399s	0.0886s	0.362737s
	RF	25.268073s	0.648894s	0.674449s	1.373108s
	SVM	1.047044s	0.299311s	0.28089s	0.163252s
	MLP	117.5147s	43.8689s	72.0586s	27.1434s
Relief	LDA	0.184384s	0.114887s	0.127802s	1.418984s
	QDA	0.1527s	0.194511s	**0.075948s**	0.239754s
	RF	23.918714s	0.470976s	0.563532s	1.794909s
	SVM	0.688252s	0.331755s	0.411625s	0.55858s
	MLP	125.0835s	47.347633s	69.8263s	31.9692s
RF		25.635199s	0.549508s	0.932766s	1.333529s
FS-Former		3791.83s	1903.49s	2504.85s	1632.89s

Table 14: Inference time of different methods.

Feature selection	Classification	Bank Marketing	CIC-IDS2017	CIC-IDS2017 with SMOTE	SMK-CAN-187
Full data (using all features)	LDA	0.009584s	0.025359s	0.164306s	0.065275s
	QDA	0.046735s	0.027803s	0.073225s	0.049459s
	RF	1.11478s	0.031827s	0.045196s	0.116954s
	SVM	0.008384s	0.05257s	0.006613s	**0.002624s**
	MLP	0.84435s	0.30344s	0.78573s	0.200445s
SFS(Embedded)	LDA	**0.004596s**	0.012163s	0.034288s	0.079067s
	QDA	0.059244s	0.039673s	0.034836s	0.009648s
	RF	0.909971s	0.032954s	0.046807s	0.123718s
	SVM	0.003818s	0.007077s	0.003164s	0.008967s
	MLP	0.72549s	0.3104s	0.495225s	0.208475s
SBS(Embedded)	LDA	0.114316s	0.087316s	0.038063s	0.020437s
	QDA	0.037209s	0.03082s	0.018875s	0.09884s
	RF	0.854182s	0.062716s	0.041515s	0.115335s
	SVM	0.010507s	0.006991s	0.008245s	0.01182s
	MLP	0.8245s	0.31416s	0.501865s	0.19537s
Relief	LDA	0.037745s	0.031456s	0.026114s	0.173468s
	QDA	0.048556s	0.194511s	0.016998s	0.042489s
	RF	0.953205s	0.035871s	0.040469s	0.186944s
	SVM	0.008157s	**0.00308s**	**0.00279s**	0.006207s
	MLP	0.86947s	0.507795s	0.47918s	0.218005s
RF		0.923309s	0.055219s	0.061035s	0.169179s
FS-Former		16.36s	10.54s	9.65s	6.32s

We also evaluate the computational cost of feature selection methods as shown in Table 15. As you can see, for Wrapper approaches, since SFS selected from an empty feature set and its initial computational cost is low, therefore SFS is faster than SBS. And they have comparable performance. However, for both methods, Some features may not be considered for evaluation and computational costs increase extremely as the data dimension increases. For the Filter approach, we can see that the computational cost for the Relief method is not

sensitive to the data dimensions but sensitive to the data size. Similar to the Filter approach, the Embedded approach RF also isn't sensitive to the data dimensions but the data size. Even in the dataset with thousand dimensions, the computational costs for the Filter approach and Embedded approach are very low.

Table 15: Feature selection time of different methods.

Feature selection	Classification	Bank Marketing	CIC-IDS2017	CIC-IDS2017 with SMOTE	SMK-CAN-187
SFS(Embedded)	LDA	43.427183s	28.116997s	86.747041s	204.669694s
	QDA	30.256156s	6.295381s	27.246544s	385.944787s
	RF	21.627891s	27.405524s	40.619354s	2561.928541s
	SVM	**5.818627s**	10.313996s	33.481154s	209.897234s
	MLP	2977.582497s	2099.491219s	15.07156s	8229.852819s
SBS(Embedded)	LDA	59.833786s	429.674011s	225.802633s	42289.500007s
	QDA	28.245725s	418.827464s	707.253661s	20671.260154s
	RF	948.689447s	555.161445s	1742.894862s	1789.883978s
	SVM	62.160358s	774.909272s	1316.228898s	158.530155s
	MLP	1089.934801s	566.080719s	1134.088934s	1569.833265s
Relief		111.839734s	17.184612s	65.044546s	**0.951115s**
RF		169.284805s	**2.674519s**	**4.378227s**	1.581664s

Chapter 5: Conclusion and Further Work

5.1 Conclusion

In this project, we implemented a framework of feature selection methods that includes four parts: data pre-processing, feature selection, classification, and evaluation. Specifically, we have implemented 3 types of feature selection methods: filter methods, wrapper methods, and embedded methods. For the filter method, we implement the Relief method. For the wrapper method, we implement SFS and SBS. For the embedded method, we implement RF and proposed and the FS-Former method. All of the above feature selection methods are combined with 5 different classifiers that are LDA, QDA, RF, SVM, and MLP. Also, to verify the generalizability of our implemented approach, we apply the constructed combination to 3 datasets with different characteristics. In order to make the datasets fit the machine learning methods, we pre-process the data by Z-score normalization and ordinal encoding. To deal the issue of class imbalance, we also adopt SMOTE oversampling technique to one of the datasets which is CIC-IDS2017. After a comprehensive evaluation, the results and figures of merit for different combinations of methods are produced, which include accuracy (ACC), balanced accuracy (BACC), the area under curve (AUC), sensitivity (SENS), and specificity (SPEC), ROC curve, confusion matrix, and computational cost. From the results, we find that all of the implemented feature selection methods successfully select the most informative features that enable the classifier to achieve better performance and lower computational costs. From the comparison of the different feature selection methods, we found the Filter method is efficient and fast to compute. Therefore, it is very suitable for applying to high-dimensional data. However, the Filter method does not consider the relationship between feature selection and classifier which may degrade the performance. For the Wrapper method, although it considers that relationship, the computational cost increases extremely as data dimensions increase. Therefore, it is only suitable for the dataset with relatively low dimensions. For the embedded method, since it is integrated with the classifier, has good performance and computational complexity, however, only part of the classifier is embedded method, so it has poor generalization capability. In this project, we also propose a feature selection method FS-Former, according to the experiment results, our proposed method successfully filters the redundant features and irreverent features and achieves comparable results with other feature selection methods.

5.2 Future Work

Due to the limited computational resource and other factors, our projects can also be improved in future work.

(1) Since the deep learning method needs elaborate fine-tuning, it requires lots of computational resources that can find the optimal parameters setting. In future work, we will try more settings to help our methods to get better performance.

(2) Due to the limited computational resource, we crap the SMK-CAN-187 into 1000 features. However, evaluating a dataset with extremely high dimensions is still needed in future work.

(3) Deep learning has been widely used in many machine learning tasks, however, this project lacks the comparison between other feature selection methods based on deep learning, which should also be included in future work.

(4) To train a transformer-based architecture, it requires large amount of training data. Therefore, in this project, the performance of our proposed FS-Former is not fully exploited. In future work, we will try to train FS-Former with with more data.

References

[1] Huang, Q., Zhang, F. and Li, X., 2018. Machine learning in ultrasound computer-aided diagnostic systems: a survey. BioMed research international, 2018.

[2] Kaur, P., Krishan, K., Sharma, S.K. and Kanchan, T., 2020. Facial-recognition algorithms: A literature review. Medicine, Science and the Law, 60(2), pp.131-139.

[3] Crawford, M., Khoshgoftaar, T.M., Prusa, J.D., Richter, A.N. and Al Najada, H., 2015. Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(1), pp.1-24.

[4] Woolson, R.F. and Clarke, W.R., 2011. Statistical methods for the analysis of biomedical data. John Wiley & Sons.

[5] Aggarwal, C.C. and Yu, P.S., 2000, May. Finding generalized projected clusters in high dimensional spaces. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 70-81).

[6] Hastie, T., Tibshirani, R., Friedman, J.H. and Friedman, J.H., 2009. The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

[7] Colaco, S., Kumar, S., Tamang, A. and Biju, V.G., 2019. A review on feature selection algorithms. In Emerging research in computing, information, communication and applications (pp. 133-153). Springer, Singapore.

[8] Raweh, A.A., Nassef, M. and Badr, A., 2018. A hybridized feature selection and extraction approach for enhancing cancer prediction based on DNA methylation. IEEE Access, 6, pp.15212-15223.

[9] Brankovic, A., Hosseini, M. and Piroddi, L., 2018. A distributed feature selection algorithm based on distance correlation with an application to microarrays. IEEE/ACM transactions on computational biology and bioinformatics, 16(6), pp.1802-1815.

[10] Bonilla-Huerta, E., Hernandez-Montiel, A., Morales-Caporal, R. and Arjona-Lopez, M., 2015. Hybrid framework using multiple-filters and an embedded approach for an efficient selection and classification of microarray data. IEEE/ACM transactions on computational biology and bioinformatics, 13(1), pp.12-26.

[11] Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp.1157-1182.

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[13] McLachlan, G.J., 2005. Discriminant analysis and statistical pattern recognition. John Wiley & Sons.

[14] Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.

[15] Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

[16] Yu, L. and Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy. The Journal of Machine Learning Research, 5, pp.1205-1224.

[17] Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on neural networks, 5(4), pp.537-550.

[18] Kira, K. and Rendell, L.A., 1992. A practical approach to feature selection. In Machine learning proceedings 1992 (pp. 249-256). Morgan Kaufmann.

[19] Almuallim, H. and Dietterich, T.G., 1994. Learning boolean concepts in the presence of many irrelevant features. Artificial intelligence, 69(1-2), pp.279-305.

[20] Hsu, H.H., Hsieh, C.W. and Lu, M.D., 2011. Hybrid feature selection by combining filters and wrappers. Expert Systems with Applications, 38(7), pp.8144-8150.

[21] Ma, S., Song, X. and Huang, J., 2007. Supervised group Lasso with applications to microarray data analysis. BMC bioinformatics, 8(1), pp.1-17.

[22] Guyon, I., Weston, J., Barnhill, S. and Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine learning, 46(1), pp.389-422.

[23] Moro, S., Cortez, P. and Rita, P., 2014. A data-driven approach to predict the success of bank telemarketing. Decision Support Systems, 62, pp.22-31.

[24] Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1, pp.108-116.

[25] Tomek, I., 1976. Two modifications of CNN. IEEE Trans. Systems, Man and Cybernetics, 6, pp.769-772.

[26] Spira, A., Beane, J.E., Shah, V., Steiling, K., Liu, G., Schembri, F., Gilman, S., Dumas, Y.M., Calner, P., Sebastiani, P. and Sridhar, S., 2007. Airway epithelial gene expression in the

diagnostic evaluation of smokers with suspect lung cancer. Nature medicine, 13(3), pp.361-366.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[28] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[29] Zhou, H.Y., Guo, J., Zhang, Y., Yu, L., Wang, L. and Yu, Y., 2021. nnFormer: Interleaved Transformer for Volumetric Segmentation. arXiv preprint arXiv:2109.03201.

[30] Yu, J., Li, J., Yu, Z. and Huang, Q., 2019. Multimodal transformer with multi-view visual representation for image captioning. IEEE transactions on circuits and systems for video technology, 30(12), pp.4467-4480.

[31] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[32] Ba, J.L., Kiros, J.R. and Hinton, G.E., 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

[33] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp.321-357.

[34] Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

[35] Breiman, L., 1996. Bagging predictors. Machine learning, 24(2), pp.123-140.

[36] Ho, T.K., 1998. The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8), pp.832-844.

[37] Genuer, R., Poggi, J.M. and Tuleau-Malot, C., 2010. Variable selection using random forests. Pattern recognition letters, 31(14), pp.2225-2236.

Acknowledgement

It's time for acknowledgement. Looking back, I have so many thanks to express, and I would like to take this opportunity to express my heartfelt gratitude to all the supervisors, friends and family members who have supported me in my undergraduate study and this project.

Here I'd first like to say "thanks" to my supervisors who helped me a lot and provided me with many constructive suggestions. Although we can't meet face to face due to the convid-19 these years, they still answered the questions I met in this product very positively.

I would also like to thank Prof. Haofeng Li of the Chinese University of Hong Kong and Prof. Jun Ding of the McGill University (they are not the supervisor for this project), who introduced me to do research and provided me with many research opportunities.

Lastly I would to thank my parents for their support in terms of affection and financial assistance in my undergraduate and study.

Appendix

北京邮电大学 本科毕业设计（论文）任务书

Project Specification Form

Part 1 – Supervisor

论文题目 **Project Title**	A Study of Feature Selection Methods for Classification		
题目分类 **Scope**	Data Science and Artificial Intelligence	Research	Software
主要内容 **Project description**	This project will implement several techniques of feature selection applied to improve automatic classification performance. Several classification problems from a publicly available database repository corresponding to biophysical data analysis will be analyzed. Those problems could include the following medical subjects: arrhythmia; breast cancer; heart failure; and hepatitis C virus. The data consist of features extracted from electrocardiographic (ECG) signals, electroencephalographic (EEG) signals, medical images, anamnesis, etc. In some cases, a preprocessing step could be required to deal with normalization, artifact removing, and missing data. The objective of feature selection methods is to obtain a sorted list of the full set of features according to some defined criteria. From this ranking of features a smaller set of features can be obtained that allows the classification performance be improved and avoid possible overfitting of the trained classification model. In this project, a comparison of SF methods is made including Relief-based and sequential feature selection (SFS) methods. As classifiers we will consider linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and support vector machine (SVM) among others. The quality of classification results will be evaluated using different ranges of the feature ranking list and several indices such as accuracy, balanced accuracy, confusion matrix, … Besides, computational cost of the different cases of classification will be estimated.		
关键词 **Keywords**	Feature selection, Classification, Pattern recognition, Machine learning, Biophysical data analysis		
主要任务 **Main tasks**	**1** Study and selection of the datasets, feature selection, and classification methods.		
	2 Design and implementation of the procedures of preprocessing, feature selection, and classification.		
	3 Experimentation: definition of the databases; tuning and debugging of the methods; implementation of figures of merit.		
	4 Evaluation and reporting of the results.		
主要成果 **Measurable outcomes**	**1** Software of the implementation of the feature selection processing step for the different datasets evaluated (and report on the implemented methods).		
	2 Software of the implementation of the classification processing step (and report on the implemented methods).		
	3 Software for obtaining results: classification accuracy, result comparison, confusion matrices (and reports on the results).		

北京邮电大学 本科毕业设计（论文）任务书

Project Specification Form

Part 2 - Student

学院 School	International School	**专业 Programme**	**e-Commerce Engineering with Law**		
姓 Family name	Liu	**名 First Name**	Chenyu		
BUPT 学号 BUPT number	2018213029	**QM 学号 QM number**	190016069	**班级 Class**	2018215115
论文题目 Project Title	A Study of Feature Selection Methods for Classification				
论文概述 Project outline **Write about 500-800 words** **Please refer to Project Student Handbook section 3.2**	Classification is one of the important tasks in machine learning which classifies each object in the data set into corresponding classes based on its features. However, An object has many features, which will cause many problems that hinder the performance of the machine learning algorithm, for example, Curse of Dimensionality and over-fitting. Therefore, reducing data dimension is considered to be an essential method in handling high-dimensional data and one of the methods to reduce data dimension is feature selection, which selects a subset of the whole features to maximize the performance and minimize the number of features of machine learning algorithm [1]. Features in an object can be roughly divided into three main types: 1. Relevant features: It is helpful for the machine learning algorithm and can improve the performance of the algorithm; 2. Irrelevant features: It is not helpful for the machine learning algorithm and will not bring any improvement to the algorithm performance; 3. Redundant features: It doesn't bring new information to our algorithm, or the information of this feature can be inferred from other features; Hence, relevant features need to be selected from all features in feature selection. There are three methods for feature selection in classification: 1. Filter Methods: Feature selection is performed first, and then the learner is trained, so the process of feature selection is independent of the learner. It is equivalent to filtering features first and then training classifiers with feature subsets. 2. Wrapper Methods: The last classifier is directly used as the evaluation function of feature selection and the optimal feature subset is selected for a specific classifier. 3. Embedded Methods: The process of feature selection is combined with the process of classifier learning and feature selection is carried out in the process of learning In this project, a comparison of SF methods is made including Relief-based and sequential feature selection (SFS) methods will be made over different datasets such as Intrusion Detection Evaluation Dataset (CIC-IDS2017), which is available at https://www.unb.ca/cic/datasets/ids-2017.html This dataset consists of network traffic data that are collected during several days which contains benign and the most up-to-date common attacks. These two categories are the classes for a classification procedure. Specifically, the data we				

	will use corresponding to the day "Thursday, July 6, 2017, Morning" consists of three kinds of web attacks: (i) Web Attack – Brute Force; (ii) Web Attack – XSS; and (iii) Web Attack – Sql Injection. In the dataset, there are over 170367 objects and 84 features for each object which contains a variety of data type including string and number. As for ReliefF and sequential feature selection (SFS) methods, they are described as below: 1. ReliefF method: It is a Filter Methods. Relief selects features that are different from different groups and are the same with similar groups. It randomly selects a sample from the training set every time, and near Hits of the sample are selected among the samples of the same class, and near Miss of the sample are selected among the samples of the different class, and then the weight of each feature is updated. The equation of this procedure is [3]: $W_I = W_{I-1} - (X_I - nearHit_i)^2 + (X_I - nearMiss_i)^2$ However, the Relief method can only deal with two-class classification problems. Thus, the ReliefF method extends the Relief method so that it can be used in multiple class problems. 2. Sequential feature selection (SFS) method It is a wrapper method. The misclassification rate was used as an evaluation metric. The search starts from the empty set, and one feature is added to the feature subset each time to make the performance reach the optimal value. If the candidate feature subset is inferior to the feature subset of the previous round, the iteration is stopped and the feature subset of the last round is taken as the optimal feature selection result. In this project, the classifier we will use includes linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and support vector machine (SVM) among others. In order to evaluate the classification results, several metrics will be used, such as accuracy, balanced accuracy, confusion matrix, recall, precision, F1-score and AUC value. The algorithm in this project will be developed using Matlab. [1] Colaco, Savina, et al. "A review on feature selection algorithms." Emerging research in computing, information, communication and applications. Springer, Singapore, 2019. 133-153. [2] Yu, Lei, and Huan Liu. "Efficient feature selection via analysis of relevance and redundancy." The Journal of Machine Learning Research 5 (2004): 1205-1224. [3] Bolón-Canedo, Verónica, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. Feature selection for high-dimensional data. Cham: Springer International Publishing, 2015.
道德规范 **Ethics**	Please confirm that you have discussed ethical issues with your Supervisor using the ethics checklist (Project Handbook Appendix 1). YES

	Summary of ethical issues: (put N/A if not applicable) N/A
中期目标 **Mid-term target.** **It must be tangible outcomes, E.g. software, hardware or simulation.** **It will be assessed at the mid-term oral.**	Software of the implementation of the data pre-processing(Task 2.1) feature selection processing step (including Sequential feature selection (SFS) method and ReliefF method)(Task 2.2) and classification processing step (including linear discriminant analysis (LDA), quadratic discriminant analysis (QDA))(Task 2.3) for CIC-IDS2017 dataset. After implementation, I will tune and debug the methods until it reaches desirable results (Task 3.2). Finally, the performance of methods will be evaluated (including accuracy, confusion matrix) for CIC-IDS2017 dataset (Task 4.1).

3

Work Plan (Gantt Chart)

Fill in the sub-tasks and insert a letter X in the cells to show the extent of each task

	Nov 1-15	Nov 16-30	Dec 1-15	Dec 16-31	Jan 1-15	Jan 16-31	Feb 1-15	Feb 16-28	Mar 1-15	Mar 16-31	Apr 1-15	Apr 16-30
Task 1 [Study and selection of the datasets, feature selection, and classification methods.]												
1.1 Study the Matlab basics	X	X										
1.2 Read papers and source code about feature selection	X	X										
1.3 Read papers and source code about machine learning classification methods		X	X	X								
1.4 Read papers to choose several datasets			X	X								
Task 2 [Design and implementation of the procedures of preprocessing, feature selection, and classification]												
2.1 Data pre-processing				X								
2.2 Implement feature selection method					X	X						
2.3 Implement machine learning and deep learning classification method					X	X	X	X				
2.4 Implement feature selection by Deep learning method transformer									X	X		
Task 3 [Experimentation: definition of the databases; tuning and debugging of the methods; implementation of figures of merit.]												
3.1 Adopt the feature selection and classification to different datasets						X	X	X				
3.2 Tuning and debugging of the methods						X	X	X				
3.3 Implementation of figures of merit							X	X				
3.4 Combine different feature selection methods with classification methods that are implemented										X	X	
Task 4 [Evaluation and reporting of the results.]												
4.1 Performance Evaluation								X	X			
4.2 Analyse the results among different methods									X			
4.3 Computational cost of the different cases of classification will be estimated									X			
4.4 Compare traditional feature selection method and transformer based method											X	X

4

北京邮电大学 本科毕业设计（论文）初期进度报告

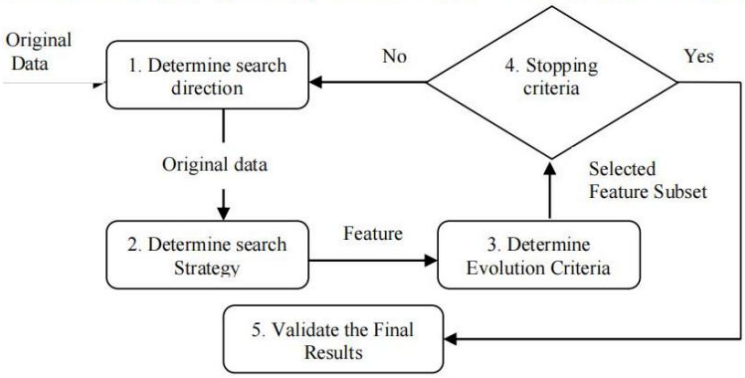
Project Early-term Progress Report

学院 **School**	International School	专业 **Programme**	**e-Commerce Engineering with Law**		
姓 **Family name**	Liu	名 **First Name**	Chenyu		
BUPT 学号 BUPT number	2018213029	**QM 学号 QM number**	190016069	班级 **Class**	2018215115
论文题目 **Project Title**	A Study of Feature Selection Methods for Classification				

已完成工作 **Finished work:**

Survey about feature selection method

Feature selection plays a important role to overcome the problem of the Curse of Dimentionality. In feature selection, the process is also very important, whose decision will affect the performance of feature selection method significantly. It includes 5 process which shown and illustrate below [1].



1. Search direction.

There are 3 types of search direction, which are forward search, backward search, and random search.

Forward search: start from an empty set, and new features are added iteratively.

Backward search: start from a full set, and features are removed iteratively.

Random search: builds the feature subset by both adding and removing of the features iteratively

2. Search strategy.

This can be divide into 3 types.

Exponential search: also known as exhaustive search, which it requires 2^N combinations of feature selection for N features and it is also a NP-hard problem. To overcome the high computational complexity, randomized search is introduced.

Randomized search: also known as heuristic search, which search features according to the predefined rules.

Sequential search: also known as greedy hill climbing, sequentially features are added to an empty set or remove features from the complete set. The issue is that the removed features will not be considered in the next iterations.

3. Evaluation criteria.

In feature selection, the most representative features are selected based on evaluation criteria, which concludes following three types.

1. Filter Methods: Feature selection is performed first, and then the learner is trained, so the process of feature selection is independent of the learner. It is equivalent to filtering features first and then

training classifiers with feature subsets.
2. Wrapper Methods: The last classifier is directly used as the evaluation function of feature selection and the optimal feature subset is selected for a specific classifier.
3. Embedded Methods: The process of feature selection is combined with the process of classifier learning and feature selection is carried out in the process of learning.

4. Stopping criteria.

The most common stopping criteria concludes:
(1) Number of features.
(2) Number of iterations.
(3) The improvement over last iteration.
(4) Evaluation performance.
In our project, we will try different process decision according to the progress of our project. Then we illustrate different types of algorithms.

Survey on dataset

Nowadays, feature selection method has been regarded as one of the most effective method to deal with the high dimension data to overcome the curse of dimensionality. Here we will introduce 3 types of application used in our project.

1. Intrusion detection

Today, web-based technologies are growing rapidly, and so are attacks against them. To solve this problem, a secure intrusion detection system (IDS) must be established. These intrusion detection systems need to deal with high-dimensional data packets containing noisy, redundant and irrelevant data.This reduces the intrusion detection rate and increases the computation time. Therefore, in order to achieve high detection rate, FS method is needed.
In this project the dataset we use is CIC-IDS2017, which consists of network traffic data that are collected during several days which contains benign and the most up-to-date common attacks.
The characteristic of this dataset is the class is not balanced and only few data are “benign” in this dataset.

2. Microarray data

This kinds of data consists of great number of features comparing with instance. In this project, the dataset we use is SMK-CAN-187 [2], which available at
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4115
In this dataset, RNA was obtained from histologically normal bronchial epithelium of smokers during time of clinical bronchoscopy from relatively accessible airway tissue. Gene expression data from smokers with lung cancer was compared with samples from smokers without lung cancer. This allowed us to generate a diagnostic gene expression profile that could distinguish the two classes. This profile could provide additional clinical benefit in diagnosing cancer amongst smokers with suspect lung cancer.
Therefore, it is a two class classification problem, which 79 total arrays run on total RNA obtained from Bronchial Epithelium of Smokers with Lung Cancer and 73 total arrays run on total RNA obtained from Bronchial Epithelium of Smokers without Lung Cancer. And there are 19993 features.

3. Text data

This kinds of data consists of a article with its words, the objective is to classify the instance to different types. In our project, the dataset we use is BASEHOCK, which is a two balanced class classification task and there are 1993 instances and 4862 features. The dataset is available at https://jundongl.github.io/scikit-feature/OLD/datasets_old.html

Data Processing

Raw features may have the many problems, which can be solved by following methods:
1. Dimensionless: that is, features of different sizes cannot be compared together. Dimensionless can solve this problem by Standardization and interval scaling. The premise of standardization is that the eigenvalues obey the normal distribution, and after standardization, they are converted into the standard normal distribution. Interval scaling method uses boundary value information to scale

the value range of features to a certain range of features, such as [0, 1].
2. Binarization of quantitative features
3. One-hot encoding for qualitative features
4. Supplementary missing value

Survey of Classification

In this project, I will also implement random forest and SVM as the classification methods as shown below.

Random forest

Random forest Is a classification algorithm proposed by Leo Breiman (2001) [3]. Through bootstrap resampling technology, it repeatedly and randomly selects N samples from the original training sample set N to generate a new training sample set training decision tree, and then generates M decision trees according to the above steps to form a random forest. The classification results of new data are determined by the number of votes in the classification tree. Its essence is an improvement of decision tree algorithm, which merges multiple decision trees together, and the establishment of each tree depends on the independent extraction of samples.

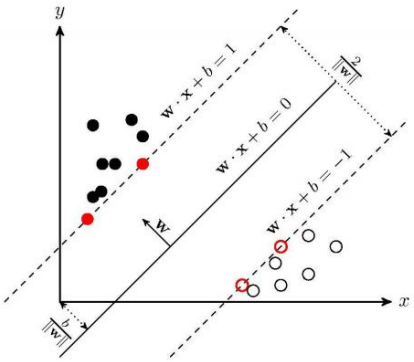
The classification ability of a single tree may be small, but after a large number of decision trees are randomly generated, a test sample can statistically select the most likely classification through the classification results of each tree.

The general process of random forest is as follows:

1) N samples were selected from the sample set with random sampling;
2) Randomly select K features from all features, and use these features to build a decision tree (generally CART, but also other or mixed) for the selected samples;
3) Repeat the above two steps m times, that is, generate M decision trees and form random forest;
4) For the new data, each tree makes a decision and finally votes to confirm the classification.

Support Vector Machines

Support Vector Machines (SVM) is a binary classification model. Its basic model is a linear classifier defined in feature space with the largest interval, which distinguishes it from perceptron. SVM also includes nuclear tricks, which makes it a substantially nonlinear classifier. The learning strategy of SVM is interval maximization, which can be formalized as a problem of solving convex quadratic programming, and is equivalent to the minimization problem of regularized hinges loss function. The learning algorithm of SVM is the optimization algorithm for solving convex quadratic programming.



As shown below, wx + b = 0 is the classification hyperplane.

Current Progress

I have implemented SFS feature selection method on CIC-IDS2017 dataset via LDA and QDA.

1. LDA

3

```
data = readtable ('Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv');
XTRAIN = table2array(data(1:160000, 8:84));
ytrain = data(1:160000, 85);
XTEST = table2array(data(160000:170000, 8:84));

ytest = data(160000:170000, 85);
ytrain = table2array(ytrain);
ytest = table2array(ytest);
ytrain =  strcmp(ytrain, 'BENIGN');
ytest =  strcmp(ytest, 'BENIGN');

MdlLinear = fitcdiscr(XTRAIN, ytrain, 'DiscrimType', 'pseudoLinear');
ypred = predict(MdlLinear, XTEST);
acc = mean(ytest == ypred);
```

The accuracy of LDA without feature selection is 0.9677

2. LDA with sequentialfs

```
data = readtable ('Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv');
XTRAIN = table2array(data(1:160000, 8:84));
ytrain = data(1:160000, 85);
XTEST = table2array(data(160000:170000, 8:84));

ytest = data(160000:170000, 85);
ytrain = table2array(ytrain);
ytest = table2array(ytest);
ytrain =  strcmp(ytrain, 'BENIGN');
ytest =  strcmp(ytest, 'BENIGN');

err = errorfun(XTRAIN, ytrain, XTEST, ytest);
%opts = statset('display', 'iter');
selected = sequentialfs(@errorfun,  XTRAIN,  ytrain);

function err = errorfun(XTRAIN, ytrain, XTEST, ytest)
MdlLinear = fitcdiscr(XTRAIN, ytrain, 'DiscrimType', 'pseudoLinear');
ypred = predict(MdlLinear, XTEST);
err = mean(ytest ~= ypred);
end
```

It starts from a empty feature set and add more features iterately, until it finds the best performance.
In our program, it only finds the first feature and the accuracy on test set is 1.0.

3. QDA

```
data = readtable ('Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv')
XTRAIN = table2array(data(1:160000, 8:84));
ytrain = data(1:160000, 85);
XTEST = table2array(data(160000:170000, 8:84));

ytest = data(160000:170000, 85);
ytrain = table2array(ytrain);
ytest = table2array(ytest);
ytrain =  strcmp(ytrain, 'BENIGN');
ytest =  strcmp(ytest, 'BENIGN');

MdlQuadratic = fitcdiscr(XTRAIN, ytrain, 'DiscrimType', 'pseudoQuadratic');
ypred = predict(MdlQuadratic, XTEST);
acc = mean(ytest == ypred);
```

The accuracy of QDA without feature selection is 0.9347

4. QDA with sequentialfs

4

```
data = readtable ('Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv')
XTRAIN = table2array(data(1:160000,8:84));
ytrain = data(1:160000,85);
XTEST = table2array(data(160000:170000,8:84));

ytest = data(160000:170000,85);
ytrain = table2array(ytrain);
ytest = table2array(ytest);
ytrain =  strcmp(ytrain,'BENIGN');
ytest =  strcmp(ytest,'BENIGN');

err = errorfun(XTRAIN,ytrain,XTEST,ytest);
%opts = statset('display','iter');
selected = sequentialfs(@errorfun, XTRAIN, ytrain);

function err = errorfun(XTRAIN,ytrain,XTEST,ytest)
MdlQuadratic = fitcdiscr(XTRAIN,ytrain,'DiscrimType','pseudoQuadratic');
ypred = predict(MdlQuadratic,XTEST);
err = mean(ytest ~= ypred);
end
```

It also only finds the first feature, and its accuracy is 0.975.

Reference

[1] Venkatesh, B., and J. Anuradha. "A review of feature selection and its methods." Cybernetics and Information Technologies 19.1 (2019): 3-26.
[2] Spira A, Beane JE, Shah V, Steiling K et al. Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. Nat Med 2007 Mar;13(3):361-6. PMID: 17334370
[3] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

是否符合进度？ On schedule as per GANTT chart?
YES

下一步 Next steps:

Implement Deep forest and SVM on different dataset.
Implement 3 different types of feature selection methods, that is, Filter Methods, Wrapper Methods and Embedded Methods.
Implementation of figures of merit.
Evaluate the methods that are implemented.

北京邮电大学 本科毕业设计（论文）中期进度报告

Project Mid-term Progress Report

学院 **School**	International School	专业 **Programme**	**e-Commerce Engineering with Law**		
姓 **Family name**	Liu	名 **First Name**	Chenyu		
BUPT 学号 BUPT number	2018213029	**QM 学号 QM number**	190016069	班级 **Class**	2018215115
论文题目 **Project Title**	A Study of Feature Selection Methods for Classification				

是否完成任务书中所定的中期目标？Targets met (as set in the Specification)?
YES

已完成工作 Finished work:

Task 1.1 Study the Matlab basics
I have studied the Matlab basics.

Task 1.2 Read papers and source code about feature selection
After investigation, I have selected 3 different types of feature selection methods. The criterion is:
1. **Filter Methods**: Feature selection is performed first, and then the learner is trained, so the process of feature selection is independent of the learner. It is equivalent to filtering features first and then training classifiers with feature subsets. For this method, we choose the **ReliefF** method [1], which selects features that are different from different groups and are the same with similar groups.

2. **Wrapper Methods:** The last classifier is directly used as the evaluation function of feature selection and the optimal feature subset is selected for a specific classifier. For this method, we choose **Sequential forward selection (SFS) and Sequential backward selection (SBS) [2]**, which sequentially select or eliminate the feature set, respectively.

3. **Embedded Methods:** The process of feature selection is combined with the process of classifier learning and feature selection is carried out in the process of learning. For this method, we use **Random Forest (RF) [3]**, which will be discussed in Task 2.2.

Task 1.3 Read papers and source code about machine learning classification methods
I have chosen 4 classification methods **linear discriminant analysis (LDA)[4], quadratic discriminant analysis (QDA)[4], Random Forest, and support vector machine (SVM)[5].** which will be discussed below.

Task 1.4 Read papers to choose several datasets
We have selected 3 types of the dataset for our model.
1. Bank Marketing Dataset [1], Prediction of client behavior.
It is a **binary** classification dataset. There are **41188** observations and **20** both **categorical and numerical** features. The class of the dataset is **balanced.** The classification goal is to predict if the client will subscribe to a term deposit.

2. SMK-CAN-187 [2], cancer diagnosing.
It is a **binary** classification dataset. There are **152** observations and **19993** both **categorical and numerical** features. The class of the dataset is **balanced.** The classification goal is for diagnosing cancer. For this dataset, we only pick up **1000** features as it is impractical for our computer to deal with **19993** features.
.
3. CIC-IDS2017, Intrusion detection.

It is a **binary** classification dataset. There are **100000** observations and **84** both **categorical and numerical** features in our experiment. The class of the dataset is **imbalanced.** The classification goal is for Intrusion detection.

Task 2.1 Data pre-processing

Raw features may have the many problems, therefore, we preprocess our data with following code.

```
new = [];

for i = 1:84

    if iscell(data{:,i})

        [GN, ~, G] = unique(data(:,i));

        new = [new, G.'];

    else

        G = zscore(table2array(data(:,i))');

        new = [new, G.'];

    end

end

ytrain = table2array(ytrain);
```

In detail, the code will judge whether the feature is a categorical feature or a numeric feature. If it is a categorical feature, it encodes the feature to an integer number in which each categorical feature corresponds to one number.

If it is numeric data, it normalizes the data by **zscore**. The mean value of the processed data is 0 and the standard deviation is 1. The equation is:

$$x^* = \frac{x - \mu}{\sigma}$$

Where μ is the mean value of the original data, and σ is the standard deviation of the original data.

Task 2.2 Implement feature selection method

SFS

This is implemented with code:

```
selected = sequentialfs(@errorfun, xtrain, ytrain,'options',opts);
```

SBS

This is implemented with code:

```
selected = sequentialfs(@errorfun, xtrain,
ytrain,'options',opts,'direction','backward');
```

Both the above methods have a **errorfun** to evaluate the feature set:

```
function err = errorfun(xtrain,ytrain,xtest,ytest)

MdlLinear = fitcdiscr(xtrain,ytrain,'DiscrimType','diagQuadratic');

% MdlLinear = fitcdiscr(xtrain,ytrain,'DiscrimType','Quadratic');

ypred = predict(MdlLinear,xtest);

[acc,sens,spec,pre,rec,F1,baacc,tp,fn,fp,tn] = metric(ypred, ytest);

err = 1-((sens+spec)/2);

end
```

ReliefF

We implement this method with following code:

```
obX = xtrain(:,:);

obY = ytrain(:,1);

tic

[selected_idx,weights] = relieff(obX,obY,100);

toc

bar(weights)

xlabel('Predictor rank')

ylabel('Predictor importance weight')


idx = weights>0.01;

xtrain = xtrain(:,idx);

xtest = xtest(:,idx);
```

The function returns selected_idx, which contains the indices of the most important predictors, and weights, which contains the weights of the predictors. Here we do feature selection by eliminating the feature with importance lower than 0.01.

Random Forest

Random forest is a classifier containing multiple decision trees, and its output categories are determined by the mode of the categories output by individual trees. Random forests has an important feature: the ability to calculate the importance of individual characteristic variables. And this feature can be applied in many aspects, for example, in the bank loan business, whether the credit of an enterprise can be correctly evaluated is related to whether the loan can be effectively recovered. However, there are many data features in the credit evaluation model, some of which are noisy, so it is necessary to calculate the importance of each feature and conduct a ranking of these features, and then select the most important feature from all the features.

The importance of a feature X in the random forest can be calculated as follows:

1: For each decision tree in the random forest, the corresponding out-of-bag (OOB, two independent sets are created in random forest. One set, the bootstrap sample, is the data chosen to be "in-the-bag" by sampling with replacement. The out-of-bag set is all data not chosen in the sampling process.) data is used to calculate its OOB data error, denoted as **OOBerr1**.

2: Randomly add noise interference to feature xi of all samples of OOB data (so that the value of samples at feature x can be randomly changed), and calculate its OOB data error again, denoted as **OOBerr2**.

3: Assuming that there are **N** trees in the random forest, the feature importance is:

$$RI = \sum_{xi}^{X} (OOBerr1 - OOBerr2)/N$$

The reason why this expression can be used as the measure of the importance of the corresponding feature is that: If a feature is randomly added with noise, the out-of-bag accuracy is greatly reduced, indicating that this feature has a great influence on the sample classification results, that is to say, its importance is relatively high.

The codes are as follows:

```
rng('default') % For reproducibility

t = templateTree('Reproducible',true); % For reproducibiliy of random predictor selections

Mdl = fitcensemble(xtrain,ytrain,'Method','Bag','NumLearningCycles',50,'Learners',t);

imp = oobPermutedPredictorImportance(Mdl);
```

3

Then it returns a vector **imp** which contains feature importance estimates. Larger values indicate predictors that have a greater influence on predictions.

Tasks 2.3 Implement machine learning and deep learning classification method

Here we introduce the classification methods used in this paper. Since our study focuses on feature selection, we will describe the classification methods briefly.

LDA and QDA

The method of LDA is simple: given the training sample method, the sample is projected to a single linear line so that the projection points of the same sample can be as close as possible and the projection points of different samples can be far away, therefore, our optimization objective is:

$$S_\omega = \Sigma 0 + \Sigma 1 = \sum_{x \in X0} (x - \mu_0)(x - \mu_0)^T + \sum_{x \in X1} (x - \mu_1)(x - \mu_1)^T$$

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$$

$$\underset{\omega}{\operatorname{argmax}} J(\omega) = \frac{\omega^T S_b \omega}{\omega^T S_\omega \omega}$$

Where S_ω and S_b are Intra-class divergence matrix and Inter-class divergence matrix respectively, and X_0 and X_1 are samples belong to different classes with the mean μ_0 and μ_1 respectively. $\Sigma 0$ and $\Sigma 1$ are class covariances. In LDA, $\Sigma 0$ and $\Sigma 1$ are assumed to be identical.

In the classification of the new sample, it is projected on the same line, and then according to the position of the projection point to determine the class of the sample.

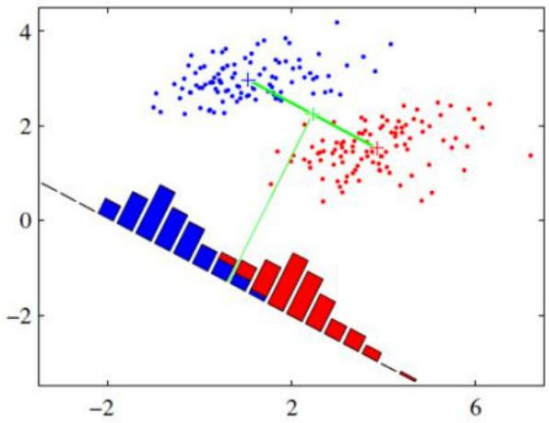


Figure 1. The illustration of LDA.

Quadratic discriminant analysis (QDA) is a variant of LDA (without assumption of $\Sigma 0 = \Sigma 1$) that allows for non-linear separation of data as shown in Figure 2.

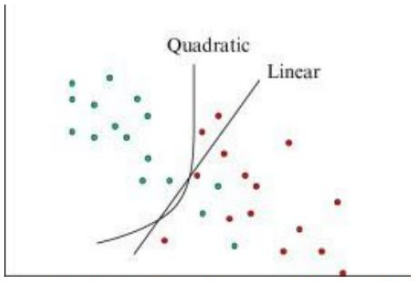


Figure 2. The difference between LDA and QDA.

We have implemented LDA and QDA via the following code:

```
MdlLinear = fitcdiscr(xtrain,ytrain);

MdlLinear = fitcdiscr(xtrain,ytrain,'DiscrimType','Quadratic');
```

SVM

Support Vector Machines (SVM) is a binary classification model. Its basic model is a linear classifier defined in feature space with the largest interval, which distinguishes it from perceptron. SVM also includes some implementations, which makes it a substantially nonlinear classifier. The learning strategy of SVM is interval maximization, which can be formalized as a problem of solving convex quadratic programming, and is equivalent to the minimization problem of the regularized hinges loss function. The learning algorithm of SVM is the optimization algorithm for solving convex quadratic programming.

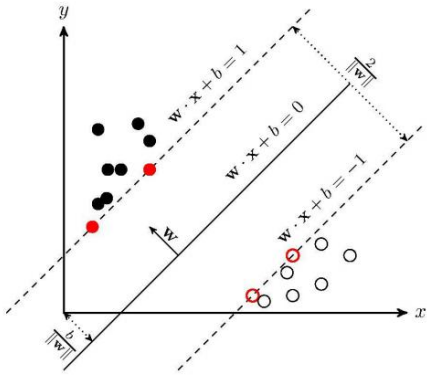


Figure 3. The illustration of LDA.

As shown below, wx + b = 0 is the classification hyperplane.

Here we implement the SVM via the following code:

```
MdlLinear = fitcsvm(xtrain,ytrain,'KernelFunction','linear');
```

Random Forest

We also use the random forest as classifier, which has been discussed above.

Proposed globformer

We also propose a transformer-based deep learning method, namely globformer, and implemented it through pytorch.

5

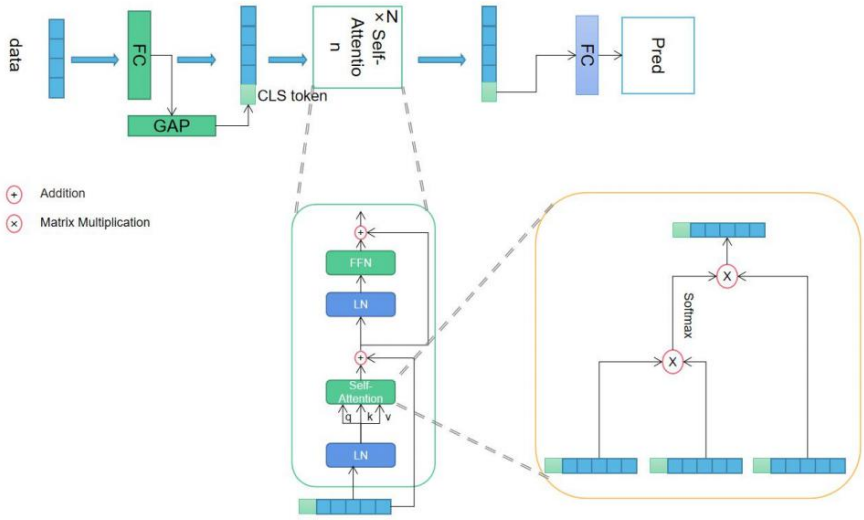


Figure 4. Architecture of globformer.

It first takes the data as input and passes through the FC (fully connected) layer for embedding. Then it adopt Global average pooling (it takes a n dimensional vector as input and output a scalar value by averaging the vector) to extract abstract global information for all data. The resulting outcome will go through N self-attention block. The self-attention block consists of alternating layers of self-attention (an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence) and FFN (Feed Forward Network) blocks. Layernorm (LN) is applied before every block and residual connections after every block. The FFN contains two FC layers with a GELU non-linearity.

Task 3.1 Adopt the feature selection and classification to different datasets

I have adopted the feature selection and classification to different datasets, the dataset description is in Task 1.4 and results are shown in Task 4.1.

Task 3.2 Tuning and debugging of the methods

There are two methods need to be tuned that are RF and Relief.

RF

We can tune the parameter **NumLearningCycles**, which is the classification trees included in the random forest. Here we select this parameter from (1,100) to different dataset. The tuning figure for each dataset shown as follows:

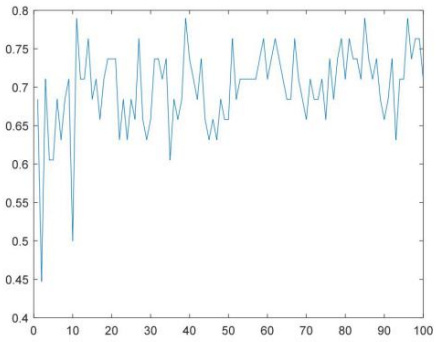


Figure 5. NumLearningCycles tuning in Bank Marketing.

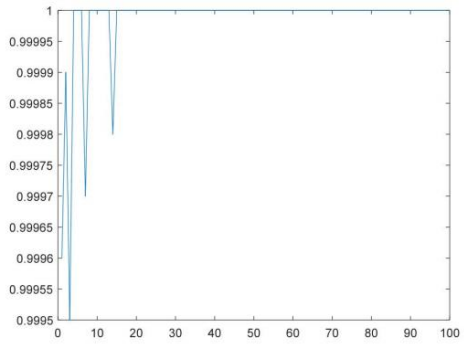


Figure 6. NumLearningCycles tuning in CIC-IDS2017.

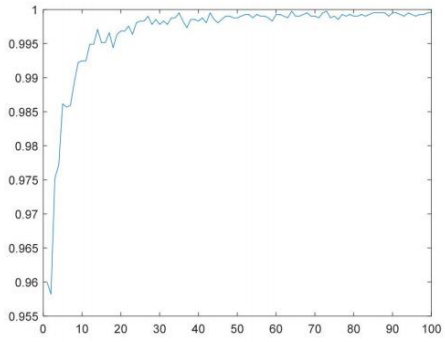


Figure 7. NumLearningCycles tuning in SMK-CAN-187.

Relieff

We also tune the Relieff by searching for the best parameter **Number of nearest neighbors,** the searching space is from 10 to 100, with a step 10. The tuning figure for each dataset shown as follows:

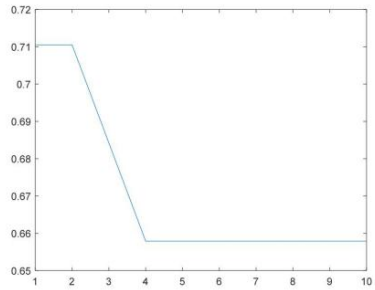


Figure 8. Number of nearest neighbors tuning in Bank Marketing.

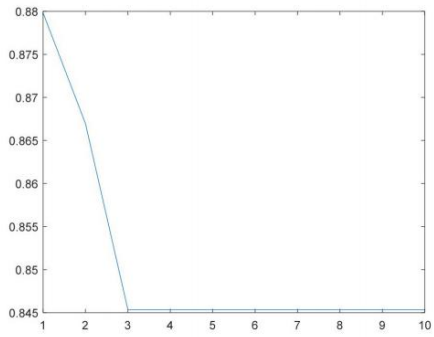


Figure 9. Number of nearest neighbors tuning in CIC-IDS2017.

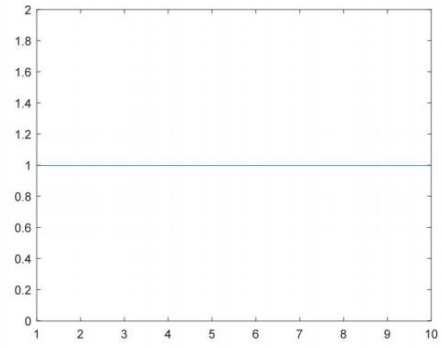


Figure 10. Number of nearest neighbors tuning in SMK-CAN-187.

Task 4.1 Performance Evaluation

To evaluate the performance of different methods and address the data imbalance problem, we introduce balanced accuracy.

$$Balanced\ error\ =\ 1-\frac{SEN+SPE}{2}$$

Where SEN is the ratio of positive classes are correctly classified and SPE is the ratio of negative classes are correctly classified.

We first adopt the methods to differet datasets.

For Relief, we eliminate the feature importance below 0.1.

8

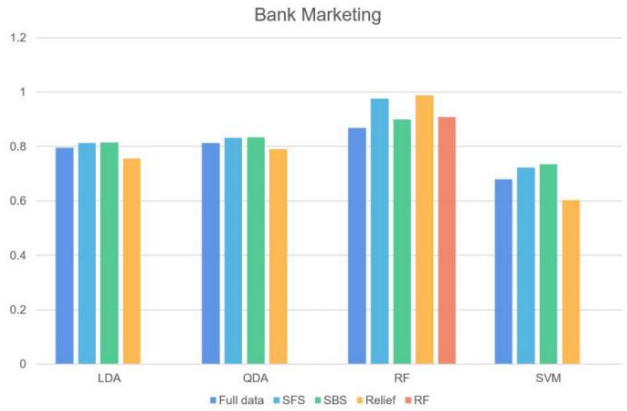


Figure 11. Performance evaluation in Bank Marketing

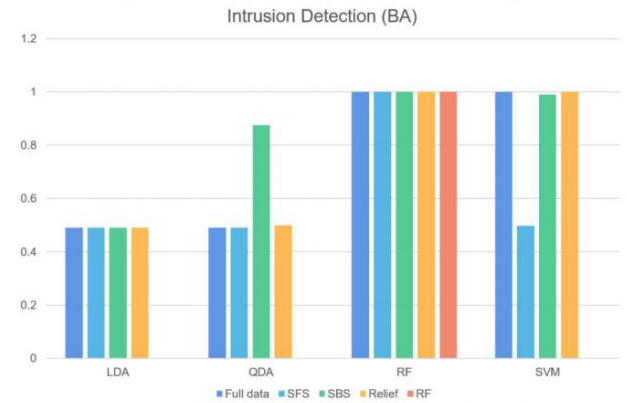


Figure 12. Performance evaluation in CIC-IDS2017.

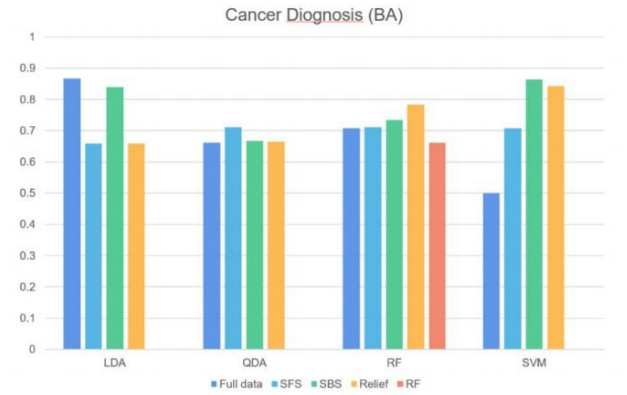


Figure 13. Performance evaluation in SMK-CAN-187.

Task 4.2 Analyse the results among different methods
Bank Marketing
For Bank Marketing, we can see that most of the classification methods can benefit from feature selection, while for LDA, QDA, and SVM, they are got enhanced via wrapper method SFS and SBS. While for RF, the performance with Relief is the best. Among them, we found that for two linear models LDA and SVM, the results with the Relief method were unsatisfactory.
We also want to know which features are selected among different feature selection methods. Here we use RF as our classification algorithm.
After merging the selected feature set with SBS, Relief, and RF itself.
We found **1 3 8 9 10 11 14 15 16 17th** features are selected by all methods. We also figure the feature importance with respect to Relief and RF.

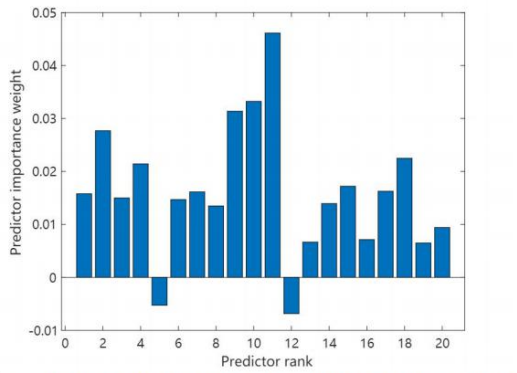


Figure 14. Importance produced by Relief in bank marketing

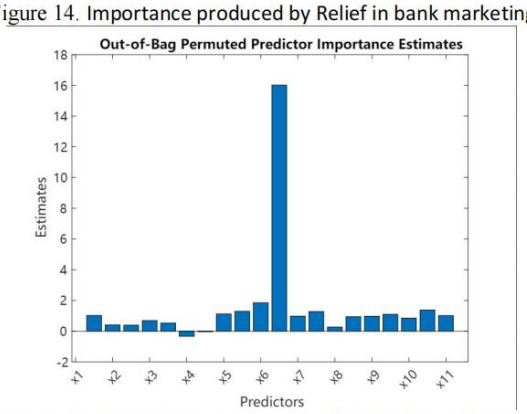


Figure 15. Importance produced by RF in bank marketing

We find that both methods treat 11th feature as the most informative.
For **intrusion detection**, we find that **2 64th** features are selected by all methods. We also figure the feature importance with respect to Relief and RF.

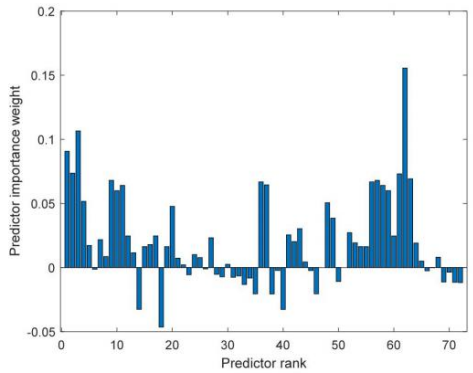


Figure 16. Importance produced by Relief in CIC-IDS2017.

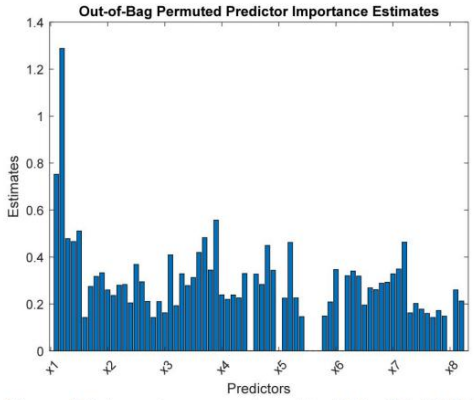


Figure 17. Importance produced by RF in CIC-IDS2017

For **cancer diagosis**, we find that **20 44 51 57 91 116 117 140 200 248 284 323 330 386 409 419 421 469 495 506 516 520 524 546 551 555 616 618 620 628 682 688 711 738 744 759 769 779 791 852 865 867 886 936 955 963 972 997th** features are selected by all methods. We also figure the feature importance with respect to Relief and RF.

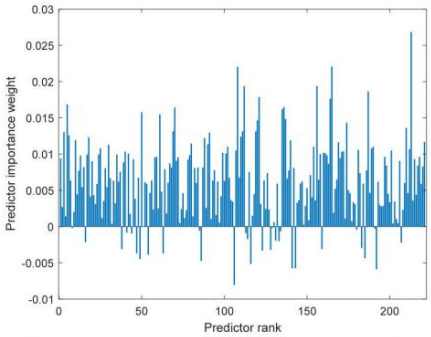


Figure 18. Importance produced by Relief in SMK-CAN-187.

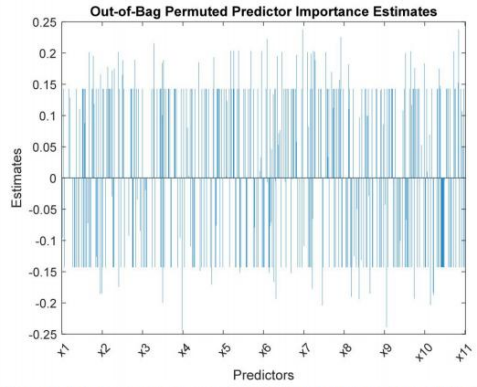


Figure 19. Importance produced by RF in SMK-CAN-187.

Task 4.3 Computational cost of the different cases of classification will be estimated

We also evaluate the Training cost of the different cases of classification.

	Full data	SFS	SBS	Relief	RF
LDA	0.049569s	0.034544s	0.04325s	0.046698s	
QDA	0.049513s	0.035544s	0.03987s	0.041148s	
RF	6.593882s	5.062407s	5.598987s	5.730348s	6.168067s
SVM	5.997366s	1.98712s	1.69602s	1.665676s	

Table 1. Training cost (seconds) on Bank Marketing.

	Full data	SFS	SBS	Relief	RF
LDA	0.555119s	0.067304s	0.320301s	0.396382s	
QDA	0.555119s	0.0852s	0.027855s	0.269136s	
RF	6.861953s	2.170376s	2.530975s	5.019605s	5.876852s
SVM	6.413194s	0.203422s	5.164808s	1.897947s	

Table 2. Training cost (seconds) on CIC-IDS2017.

	Full data	SFS	SBS	Relief	RF
LDA	2.544254s	0.209999s	0.059439s	0.012699s	
QDA	0.572072s	0.105944s	0.051069s	0.014173s	
RF	0.7083s	0.648397s	0.383576s	0.323688s	0.256239s
SVM	0.485824s	0.011496s	0.011241s	0.008818s	

Table 3. Training cost (seconds) on SMK-CAN-187.

And we evaluate the feature selection cost for RF (Since RF is also an embedded feature selection method)

	SFS	SBS	Relief	RF
RF	3659.22994s	3813.93081s	118.826706s	6.168067s

Table 4. Feature selection cost (seconds) on Bank Marketing.

	SFS	SBS	Relief	RF
RF	2395.9562s	3813.93081s	13.190827s	311.607537s

Table 5. Feature selection cost (seconds) on CIC-IDS2017.

	SFS	SBS	Relief	RF
RF	830.869913s	170.922599s	0.762486s	1.422401s

Table 6. Feature selection cost (seconds) on SMK-CAN-187.

Reference

[1] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in European Conference on Machine Learning (Springer, Berlin, 1994), pp. 171–182

[2] Colaco, Savina, et al. "A review on feature selection algorithms." Emerging research in computing, information, communication and applications. Springer, Singapore, 2019. 133-153.

[3] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[4] McLachlan, Geoffrey J. Discriminant analysis and statistical pattern recognition. John Wiley & Sons, 2005.

[5] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

[6] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

[7] Spira A, Beane JE, Shah V, Steiling K et al. Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. Nat Med 2007 Mar;13(3):361-6. PMID: 17334370

尚需完成的任务 Work to do:

Task 2.4 Implement feature selection by Deep learning method transformer

Task 4.4 Compare traditional feature selection method and transformer-based method

Implement a receiver operating characteristic (ROC) curve analysis.

A statistical significance test should be applied to determine if the results of the previous point (mean and standard deviation) are due to random effects or not.

The variability of the results should be estimated. Thus, Montecarlo experiments should be implement ed by changing the training and testing datasets randomly (including several cross validation batches) Thus, the mean and standard deviation of the results of those experiments can be estimated and discus sed.

The confusion matrices of the different classification results should be analyzed.

Add more metrics and figures of merit to evaluate the results.

This project will finished on time.

存在问题 Problems:
The deep learning algorithm needs lots of computational resource.

拟采取的办法 Solutions:
Rent a GPU server.

论文结构 Structure of the final report:

1. Specification
It provide a clear and precise description of both the problem the project will address, and the proposed solution.

2. Abstract
A short overview of the whole report.

3. Pre-reading knowledge
information for a reader with this level of technical competence to understand what have done without needing to refer to external sources.

4. Table of contents
A full table of contents is very important to allow the reader to quickly locate information in report.

5. Introduction
Introduce project to the reader.

6. Background
The background chapter should include relevant information that explains the background context of project to the reader.

7. Design and implementation
This describes the design and implementation of whatever system have produced.

8. Results and discussion
This show what the outcome of design and implementation phase was.

9. Conclusion and further work
The conclusion chapter should briefly restate what has been written in the preceding chapters.

10. References (Bibliography)
This include as much information about each reference so that the reader could find the document cited easily if they need to.

11. Acknowledgments
This should be a short section that thanks my Supervisor and any other people who helped you with my project.

12. Appendices
This contain information that think may be helpful or relevant for the reader but that is not directly relevant to the story of your project.

13. Risk and environmental impact assessment
Describe any factors that could prevent successful completion of project.

北京邮电大学 本科毕业设计（论文）教师指导记录表

Project Supervision Log

学院 School	International School	**专业 Programme**	**e-Commerce Engineering with Law**		
姓 Family name	Liu	**名 First Name**	Chenyu		
BUPT 学号 BUPT number	2018213029	**QM 学号 QM number**	190016069	**班级 Class**	2018215115
论文题目 Project Title	A Study of Feature Selection Methods for Classification				

Please record supervision log using the format below:

Date: dd-mm-yyyy
Supervision type: face-to-face meeting/online meeting/email/other (please specify)
Summary:

Date: 01-10-2021
Supervision type: email
Summary: discuss how I start my project and give me some information about project's background.

Date: 29-10-2021
Supervision type: email
Summary: provide me with a dataset and some sample codes.

Date: 06-11-2021
Supervision type: email
Summary: discuss the review of our project.

Date: 15-11-2021
Supervision type: face-to-face meeting
Summary: discuss about external project and draft specification.

Date: 18-11-2021
Supervision type: email
Summary: discussed the project specification.

Date: 15-12-2021
Supervision type: face-to-face meeting
Summary: talk about your project progress according to project plan.

Date: 20-12-2021
Supervision type: email
Summary: discussed the next step of Final Project.

Date: 09-01-2022
Supervision type: email
Summary: discussed the early term report.

1

Date: 28-01-2022
Supervision type: face-to-face meeting
Summary: talk about your project progress according to project plan.

Date: 21-02-2022
Supervision type: email
Summary: discussed the progress of the report.

Date: 28-02-2022
Supervision type: email
Summary: discussed the mid-term report.

Date: 23-03-2022
Supervision type: email
Summary: discussed the existing problem of project.

Date: 06-04-2022
Supervision type: email
Summary: discussed the the content of final report.

Date: 08-04-2022
Supervision type: face-to-face meeting
Summary: Doing mock viva and discuss final report problem.

Date: 11-04-2022
Supervision type: email
Summary: discuss and review the final report.

Date: 12-04-2022
Supervision type: face-to-face meeting
Summary: Doing mock viva and discuss viva problem.

Date: 27-04-2022
Supervision type: email
Summary: discussed final version of final report.

Date: 29-04-2022
Supervision type: face-to-face meeting
Summary: discuss the final viva.

Risk and environmental impact assessment

(1) Prevents the successful completion of the project

Description of risk	Description of Impact	Likelihood rating	Impact rating	Result R	Rating of risk	action
Insufficient computational resource	For a dataset with high dimensions, adopting machine learning requires a large computational resource	5	2	10	Significant Risk	Crap the dataset with fewer features
Deep learning methods need to collaborate fine-tuning	For deep learning methods, a proper hyper-parameter is essential to train an optimal model	4	2	8	Significant Risk	Adopt the appropriate hyper-parameter searching strategy

(2) Causes potential harm to people and /or animals

Description of risk	Description of Impact	Likelihood rating	Impact rating	Result R	Rating of risk	action
Biased feature selection	The selected feature may have a stereotype for	4	4	16	High Risk	Manually eliminate the biased features

	some persons. For example, the algorithm may select gender as the informative feature to judge criminals.					