applied sciences

MDPI

Article

Key Quality Indicators Prediction for Web Browsing with Embedded Filter Feature Selection

Su Xie [1], Ke Li [1,*], Mingming Xiao [1], Le Zhang [2] and Wanlin Li [3]

[1] College of Smart City, Beijing Union University, Beijing 100101, China; 13661365299@163.com (S.X.); xxtmingming@buu.edu.cn (M.X.)
[2] Department of CTNET2025, China Telecom Research Institute, Beijing 102209, China; zhangle@chinatelecom.cn
[3] College of Information Technology, North China University of Technology, Beijing 100041, China; wanlin.li@eversino.com
* Correspondence: like@buu.edu.cn

Received: 3 March 2020; Accepted: 15 March 2020; Published: 21 March 2020

Abstract: In this paper, the prediction of over-the-top service quality is discussed, which is a promising way for mobile network engineers to tackle service deterioration as early as possible. Currently, traditional mobile network operation often takes appropriate remedial measures, when receiving customers' complaints about service problems. With the popularity of over-the-top services, this problem has become increasingly serious. Based on the service perception data crowd-sensed from massive smartphones in the mobile network, we first investigated the application of multi-label ReliefF, a well-known method of feature selection, in determining the feature weights of the perception data and propose a unified multi-label ReliefF (UML-ReliefF) algorithm. Then a feature-weighted multi-label k-nearest neighbor (ML-kNN) algorithm is proposed for the key quality indicators (KQI) prediction, by combining the UML-ReliefF and ML-kNN together in the learning. The experimental results for web browsing service show that UML-ReliefF can effectively identify the most influential features of the data and thus, lead to better performance for KQI prediction. The experiments also show that the feature-weighted KQI prediction is superior to its unweighted counterpart, since the former takes full advantage of all the features in the learning. Although there is still much room of improvement in the precision of the prediction, the proposed method is highly potential for network engineers to find the deterioration of service quality promptly and take measures before it is too late.

Keywords: feature selection; over-the-top; key quality indicator; multi-label k-nearest neighbor; mobile network

1. Introduction

With the large-scale commercialization of long-term evolution (LTE) networks and the widespread popularity of smartphone, the over-the-top (OTT) services, which are dominated by Internet companies, have developed rapidly and are becoming the major services on the phone. On the one hand, traditional services including voice, SMS, and other value-added services, have been largely replaced. On the other hand, new services have stimulated the convergence of mobile networks and the Internet. The quality of experience (QoE) of OTT services are no longer determined solely by the quality of the mobile network itself.

For traditional operation and maintenance of mobile networks, engineers have always taken measures after receiving customer complaints. It is difficult to cope with the rapid and ever-changing network and service environments. Instead, in certain circumstance it is better to identify or even predict traffic demands, service quality and potential service degradations [1–3]. If the service quality can be predicted precisely, it will enable engineers to take appropriate measures before the service deterioration leads to complaints, or to recommend the best service to customers. QoE can then be guaranteed.

With the popularity of smartphones and the application of a large number of sensors, smartphones have gradually replaced drive test equipment, and have become important points to collect users' service perceptions [4]. The data is highly valuable not only for network operators themselves, but also for OTT service providers. Based on massive service perception data and appropriate machine learning methods, one can effectively improve the intelligence of the network operations. Gartner defined this new paradigm as Artificial Intelligence for IT Operations (AIOps) [5].

Taking the OTT web browsing as an example, here we try to employ the improved feature selection and multi-label learning algorithm to predict its key quality indicators (KQI) for the users in the mobile network, with the perceptional data crowd-sensed at the terminal end.

The rest of the paper is organized as follows: In Section 2 the related works are introduced, including mobile crowdsensing, KQI of OTT web browsing, multi-label feature selection (FS) and multi-label k-nearest neighbor (ML-kNN). In Section 3 we propose an ML-kNN based algorithm for the prediction of KQI with a modified multi-label feature selection. In Section 4, the proposed method is validated through experiments and compared with the traditional methods. Finally, the work of this paper is concluded in Section 5.

2. Related Works

The following is a brief introduction of the terminal-side service perception data acquisition and evaluation, as well as multi-label feature selection and ML-kNN.

2.1. OTT Service Perception with Mobile Crowdsensing

Recently, the crowdsourcing-based network measurement at terminal-side has received increasing attention from both the academic and industrial sectors. It is believed that more realistic service perception information can be acquired, since the data access point is closer to the user than that of the network-side measurements. Therefore, it provides a new perspective for the application of big data in the OTT service operations. In Reference [4] this new paradigm of data acquisition and analysis is named as mobile crowdsensing (MCS). It is reported to have been applied in many fields, including mobile network operation [6,7], urban transportation monitoring and daily life [8].

In this paper, the service perception data are collected with MCS method on massive number of smartphones in the live LTE network, and then utilized to evaluate the KQI of target services. The extensive span in both the temporal and spatial domains of the data enables a more realistic evaluation of the QoE. However, a major drawback of MCS is the deployment of data acquisition agent, either as stand-alone Application or Software Development Kit. Another issue is the privacy protection of the end-users.

The general process of the perceptional data acquisition are as follows. First, the agent installed on the phone monitors the user's service behavior; in case a pre-defined condition is met (e.g., the user is visiting a target URL through the browser), the data acquisition function is triggered. Finally, the data are uploaded to the server-end for further analysis.

In order to protect user privacy, generally sensitive information such as phone number, and text of applications, is not collected. Only the terminal ID is employed to differentiate the users, and is desensitized before use.

2.2. KQI for OTT Web Browsing

KQI is a collection of metrics which is utilized to characterize the quality of a service, usually based on the service characteristics and their relevance to the user experience. An OTT web browsing process generally consists of four steps: DNS resolution, TCP connection, HTTP page request and HTTP page response. Specifically, its KQIs are defined as follows [9]:

(1) First packet delay (D_k): The time interval between a page request and the arrival of the first HTTP 200 OK packet. That is,

$$D_k = T_{200} - T_{req} \tag{1}$$

where T_{req} is the time when the user initiates a page request, and T_{200} is the time when the terminal receives the first 200 OK packet.

The first packet delay can be further broken down into three sub-indicators, namely, D_{dns} (DNS resolution delay), D_{tcp} (TCP connection delay) and D_{get} (HTTP request delay):

$$D_k = D_{dns} + D_{tcp} + D_{get} \tag{2}$$

$$\begin{cases} D_{dns} = T_{dns} - T_{req} \\ D_{tcp} = T_{tcp} - T_{dns} \\ D_{get} = T_{200} - T_{tcp} \end{cases} \tag{3}$$

where T_{dns} refers to the time when the terminal receives the DNS resolution result, and T_{tcp} is the time when the terminal sends a TCP connection confirmation.

(2) Page open delay (D_p): The time duration since the user initiates the URL request until the whole page (text content only, excluding the resources in the page) is downloaded and rendered. The page open delay reflects the total time that the user needs to wait. It is defined as,

$$D_p = D_k + D_{res} \tag{4}$$

where $D_{res} = T_{res} - T_{200}$ refers to the HTTP response delay. Here, T_{res} is the time when the terminal responds the [FIN, ACK] message.

It should be noted that OTT web browsing on the phone is quite different from the traditional one on the PC side, specifically, they are as follows.

1. Different service process and mechanism: In order to reduce unnecessary throughput consumption and improve user experience, both the browser and the web pages on the phone are tailored for screen size and resolution. The downloading and rendering mechanism, which is in general on-demand based, is also different from that of its PC counterpart.
2. Different factors that affect the service experience: With the whole mobile network getting involved, the factors that affect an OTT web browsing is much more complicated than that of the traditional web service. In particular, the fast-changing wireless environment has a great impact on the service experience, thus bringing challenges to the trouble shooting of service deterioration.
3. Different definition of the indicators: for example, the DNS resolution delay includes the delay of air interface setup, which is not considered in the PC web browsing.

In addition to the KQIs, the network and the location information are also collected simultaneously. They are utilized to analyze and locate the service problems. The network information generally includes wireless ID and coverage parameters. For LTE network, for instance, the wireless ID refers to {tracking area code (TAC), eNodeB ID, cell ID (CI)}, while the coverage parameters include Reference Signal Receiving Power (RSRP), Reference Signal Receiving Quality (RSRQ), and signal-to-interference and noise ratio (SINR).

2.3. Modelling of KQI Prediction

While the user is experiencing an OTT service on his phone, QoE is often affected by various factors, including mainly the network, the service provider (SP) platform and the terminal. When the user is residing in a specific scenario, we can then predict the QoE of the service the user is requesting, by utilizing the network information the user is experiencing. It is a typical problem of classification or regression. Based on the prediction, one can either recommend appropriate services to the user, or find the potential service problems that the user may encounter and take prompt measures to solve them.

In addition, the most important research in the field of web service recommendation is QoS prediction as well. Currently, collaborative filtering (CF)-based methods [3,10,11] are commonly used. There are two types of collaborative filtering approaches, namely model-based and neighborhood-based.

In the neighborhood-based approach, the unknown values are predicted by employing the values of similar users or items. Model-based CF employs training data to find the user interest pattern and then predict the user's interest in the items that have not been accessed. But, model building in the model-based CF is very time-consuming.

In [3], an integrated QoS prediction approach, which unifies the modeling of QoS data via multi-linear-algebra based concepts of tensor, has been proposed. It enables efficient web service recommendation for mobile clients via tensor decomposition and reconstruction optimization.

A prominent problem of the above methods is that too few dimensional data are employed. For example, in [3] only three dimensions, user ID, web service and time interval, are considered for the prediction of two QoS indicators. Less information always leads to over-fitting. What's more, in most cases there is a lack of real-world QoS dataset, especially the ones containing spatial-temporal information.

Most contemporary QoS prediction methods exploit the QoS characteristics for a few specific dimensions, and do not exploit the structural relationships among the multi-dimensional QoS data in a high-dimensional scenario.

In this paper, the service perception dataset employed is acquired through MCS. It contains rich environmental information of the service, up to 13-dimension or higher. The prediction of KQIs with environmental features can be modelled as a typical classification problem. A promising prediction can be made by adapting appropriate machine learning algorithms.

Furthermore, here we consider only multi-label binary classification in our study. That is, only the positive or negative labels of KQI indices are to be predicted, unlike the above-mentioned prediction of network traffic or KQI. To mitigate the interference of low-correlated attributes on the prediction, and to avoid unnecessary computational complexity, a feature selection process is performed before the classification.

3. Feature Weighted ML-kNN for KQI Prediction

In this section, a KQI prediction algorithm is proposed, by employing ML-kNN together with an improved feature selection.

Details of the proposed algorithm are as follows.

3.1. Data Preprocessing

Generally, preprocessing is necessary since the service perception data acquired in the real network always suffer from missing attributes and noises. The preprocessing includes removal of abnormal samples, conversion and normalization, etc.

(1) Constructions of initial feature set and label set

In a mobile network (e.g., LTE), in case a user is visiting a target webpage by a web browsing APP (such as UCweb, QQ browser, etc.) on the phone, a service perception sample is then generated by the MCS agent deployed on the phone. All the samples received at the server end constitute the initial MCS dataset $\overline{\boldsymbol{D}} = \left\{ (\overline{\boldsymbol{x}}_i,\ \overline{\boldsymbol{Y}}_i) \middle| 1 \le i \le m \right\}$. Each sample contains mainly the following attributes: Date, Time, TAC, Cell ID, Longitude, Latitude, RSRP, RSRQ, Phone ID, Phone Model, Website name, Website IP, DNS IP, D_{dns}, D_{tcp}, D_{req}, and D_{res}.

Here we take a subset of the dataset field, {Date, Time, TAC, CI, Longitude, Latitude, RSRP, RSRQ, Website name, Website IP, DNS IP, Phone ID, Phone Model}, as the initial feature set $\overline{\boldsymbol{x}} = \left\{ \overline{x}_1,\ \overline{x}_2, \ldots, \overline{x}_p \right\}$, p = 13. To make it clear, here we number them with Feature IDs from 1 to 13 in sequential order. Among these the fields {Date, Time, Longitude, Latitude, RSRP, RSRQ} are numerical, and the rest are nominal.

The initial label set consists of all the KQI indicators $\{D_{dns}, D_{tcp}, D_{req},$ and $D_{res}\}$, and is represented as $\overline{\boldsymbol{Y}} = \left\{\overline{y}_1, \overline{y}_2 \ldots, \overline{y}_q\right\}, q = 4$.

(2) Normalization of the features

All the numerical-valued features are normalized as follows.

$$x_i = \frac{g(\overline{x}_i) - LB(\overline{x}_i)}{UB(\overline{x}_i) - LB(\overline{x}_i)},\ 1 \leq i \leq m \tag{5}$$

where $\overline{x}_i$ represents the initial value of the i-th feature, and $g(\overline{x}_i)$ is a truncation function with an upper bound $UB(\overline{x}_i)$ and a lower bound $LB(\overline{x}_i)$, i.e.,:

$$g(\overline{x}_i) = \begin{cases} UB(\overline{x}_i), & if\ \overline{x}_i > UB(\overline{x}_i) \\ LB(\overline{x}_i), & if\ \overline{x}_i < LB(\overline{x}_i) \\ \overline{x}_i, & otherwise \end{cases} \tag{6}$$

In practice, sampling error or individual difference among the phones often leads to extremely small or large sample values. To mitigate its influence, box plot method is employed to determine the lower and upper bounds for the truncation, namely,

$$\begin{cases} LB(\overline{x}_i) = \max(Q1(\overline{x}_i) - 3IQR(\overline{x}_i),\ \min(\overline{x}_i)) \\ UB(\overline{x}_i) = \min(Q3(\overline{x}_i) + 3IQR(\overline{x}_i),\ \max(\overline{x}_i)) \end{cases} \tag{7}$$

where $Q3(\overline{x}_i)$ and $Q1(\overline{x}_i)$ represent the upper and lower quartile of box plot of the i-th feature, respectively. $IQR(\overline{x}_i) = Q3(\overline{x}_i) - Q1(\overline{x}_i)$ refers to the interquartile range.

(3) Boolean-conversion of labels

For binary classification task, the labels in the dataset need to be first converted into Boolean values, with the predefined hard decision thresholds $\{T_1 \sim T_q\}$:

$$y_j = [\![\overline{y_j} < T_j]\!], 1 \leq j \leq q; \tag{8}$$

The function $[\![c]\!]$ indicates that when condition c holds, it returns 1, otherwise 0.

Then we get the dataset $\boldsymbol{D'} = \{(\boldsymbol{x}_i, \boldsymbol{Y}_i) | 1 \leq i \leq m\}$.

3.2. Improved ML-ReliefF Feature Selection

In order to improve the validity of the features in the training dataset for KQI prediction, feature selection is conducted after the preprocessing.

As the most well-known filter FS algorithm, ReliefF is widely used in solving multi-classification and regression problem [12]. Many scholars have done a lot of work on the optimization and application of ReliefF [13,14].

As more and more application scenarios of machine learning are multi-label classification problems, the algorithm needs to be extended to deal with multi-label data. A multi-label ReliefF (ML-ReliefF) is proposed in [15]. The algorithm assumes that the categories of multi-labeled data are co-occurring, and that the contributions of all the labels are equal. Satisfactory experimental results are observed on several public datasets by optimizing the update strategy of the feature weights. Finally, a subset of the features whose total weight is larger than a predefined threshold, or simply the top N features with the largest weight are selected for subsequent learning procedure.

Another multi-label ReliefF algorithm, namely, ReliefF for multi-label (RF-ML), is proposed in [16]. The algorithm considers the interaction among features. It defines a function that characterizes the dissimilarity of the label vector between samples (such as Hamming distance), and finds the nearest neighbor by means of the multi-label dissimilarity function.

It is found that ML-ReliefF actually converts the multi-label problem into a series of single-label problems, which does not consider similarity and the interaction of label vectors in the samples. For most scenarios however, the assumption of absolute independency among the labels is difficult to satisfy.

In view of the above-mentioned problems, a new algorithm, namely, unified multi-label ReliefF (UML-ReliefF), is proposed based on ML-ReliefF. Specifically, for a dataset with q labels, k nearest neighbor samples with t (t = 0~q) identical labels are sequentially searched according to Hamming distance in the label space and Euclidean distance in the feature space. During the iteration of the feature weight vector, the weights are updated according to the Hamming distance of label vectors between the sample and each of its nearest neighbors. The closer the label vectors between the two samples are, the heavier the punishment is for the difference in the feature space, or vice versa.

The pseudo code of the UML-ReliefF is illustrated in Algorithm 1.

Algorithm 1. Pseudo code of unified multi-label ReliefF (UML-ReliefF) algorithm.

```
$\boldsymbol{W}$ = UML-ReliefF($\boldsymbol{D'}$, $n$, $k$)
Input: training set $\boldsymbol{D'} = \{\boldsymbol{x}_1 \sim \boldsymbol{x}_m\}$, no. of iterations $n$, no. of neighbors $k$
Output: Feature Weight Vector $\boldsymbol{W}$
Initialize $\boldsymbol{W} = (w_1, w_2, \ldots, w_p)$; //$p$ refers to the dimension of feature set.
for $i$ = 1 to $n$ do
    Select randomly a sample from $\boldsymbol{D'}$ as $\boldsymbol{R}_i$, $class(\boldsymbol{R}_i)$ = $\boldsymbol{h}$ = $(h_1, h_2, \ldots, h_q)$ //$q$ is the dimension of label set.
    for $t$ = 0 to $q$ do
        Search $k$ nearest neighbors with $t$ same labels, and denote as $\boldsymbol{H}_j^t(\boldsymbol{R}_i)$, $j = 1 \sim k$;
    endfor
    for $l$ = 1 to $p$ do
        Update each weight according to Equation (9):
        $\boldsymbol{W}[l]\boldsymbol{W}[l] + \frac{1}{nk}\sum_{t=0}^{q}\sum_{j=1}^{k}(q-2t)diff\left(l, \boldsymbol{R}_i, \boldsymbol{H}_j^t(\boldsymbol{R}_i)\right)$ (9)
        //$diff\left(l, \boldsymbol{R}_i, \boldsymbol{H}_j^t(\mathrm{C})\right)$ is the distance between $\boldsymbol{R}_i$ and $\boldsymbol{H}_j^t(\boldsymbol{R}_i)$ on the $l$-th feature.
    endfor
endfor
```

Since the feature weights reflect the similarity between the two samples in the nearest neighbor searching, is it feasible to weight the distance with the instantaneous feature weights during neighbor searching? This may speed up the convergence. Specifically, the Euclidean distance can be modified to be a normalized weighted one in the search for $\boldsymbol{H}_j^t(\boldsymbol{R}_i)$. In addition, the weights are initialized to be 0 in Reference [15], which may lead to the non-normalization problem. Therefore, they shall be set to 1 instead.

3.3. KQI Prediction Based on Feature Weighted ML-kNN

ML-kNN [17,18] is the extension of classical kNN in dealing with multi-label classification. There have been many modifications proposed thereafter. For instance, F. Li proposes a granular feature weighted kNN for multi-label learning based on mutual information in Reference [19]. By granulating the label space into a plurality of label granules, the feature weights are then calculated for each label granule. The features are weighted according to the label classification knowledge embedded, and then the problems arising from the explosion of label combinations and inter-label correlation in the original ML-kNN can be mitigated.

As we know, the accuracy of KNN-based classifier and clustering analysis is largely dependent on the type of distance metric employed [20,21]. For example, in [22] the performance of kNN in ECG analysis in case of Euclidean distance, Manhattan distance and correlation distance are compared, and finally Euclidean distance performs the best. For simplicity, here Euclidean distance, one of the most widely used distance measures, is employed. Generally, in the neighbor searching of ML-kNN algorithm, the distance of samples in feature space is calculated with equal-weighted Euclidean distance. The filter feature selection is separated from the subsequent learning tasks. Nevertheless, the feature weights actually reflect how significantly each feature influences the labels. Therefore, is it possible to utilize the feature weights to intervene in the calculation of distance in the neighbor searching? It might be able to search for more effective neighbors.

Considering that the KQI prediction task involves only a few number of features and labels, the main purpose of feature selection herein is not to reduce the number of features involved in the learning, but to utilize the feature weights to improve the performance of the KQI prediction instead. Therefore, a feature-weighted ML-kNN is proposed to predict the KQI. In this case, the FS process is tightly embedded into the machine learning task that follows. The UML-ReliefF is no longer an independent process as conventional filter FS does. Thus, we call it the embedded filter feature selection.

In addition, from the perspective of practical scenario, it is usually necessary to use historical data samples of a few previous days to predict the service quality at the moment, so as to ensure the timeliness of the prediction.

Therefore, based on the preprocessing and feature selection, the KQI prediction can be performed with feature-weighted ML-kNN. The pseudo code of the algorithm is presented in Algorithm 2.

Algorithm 2. Pseudo code of feature-weighted multi-label k-nearest neighbor (ML-kNN) algorithm.

```
Input: Initial training set $\overline{\boldsymbol{D}}$, Unknown sample $\boldsymbol{x}_0$, Prediction window length $L$;
Output: Prediction of label vector $\boldsymbol{Y}_0$ of unknown sample;
1. Preprocess $\overline{\boldsymbol{D}}$ according to Section 3.1 to obtain $\boldsymbol{D}'$;
2. Obtain $\boldsymbol{W} = (w_1, w_2, \ldots, w_p)$ according to UML-ReliefF in Section 3.2; // $p$: dimension of feature set.
3. Sort the features according to their weights in descending order and retain Top $N$ features, reconstruct $\boldsymbol{D}'$ to be $\boldsymbol{D}''$ accordingly;
4. Select $\boldsymbol{x}_i$ from $\boldsymbol{D}''$ where $x_{i1}$ is in the range $(x_{01} - L,\ x_{01} - 1)$;
5. Reconstruct the training set $\boldsymbol{D} = \{(\boldsymbol{x}_i, \boldsymbol{Y}_i) \middle| 1 \le i \le m\prime\}$;
6. Identify $k$-nearest neighbors $\mathcal{N}(\boldsymbol{x}_i)$ for each $\boldsymbol{x}_i$ in $\boldsymbol{D}$, in which the distance is calculated with weighted Euclidean distance as in Equation (10):
7. $\mathrm{dis}\left(\boldsymbol{x}_i,\ \boldsymbol{x}_j\right) = \sqrt{\sum_{\mathrm{n}=1}^{N} w_n \cdot \left(\boldsymbol{x}_{in} - \boldsymbol{x}_{jn}\right)^2}$ (10)
8. for $j$ = 1 to $q$ do //$q$: dimension of label set.
    9. Calculate the prior probabilities $\mathbb{P}\left(H_j\right)$ and $\mathbb{P}\left(\overline{H}_j\right)$ according to Equation (11):
    10. $\mathbb{P}\left(H_j\right) = \frac{s + \sum_{i=1}^{m\prime} [\![ y_j \in \boldsymbol{Y}_i ]\!]}{2s + m\prime},\ \mathbb{P}\left(\overline{H}_j\right) = 1 - \mathbb{P}\left(H_j\right);$ (11)
    11. Calculate the normalized frequency arrays $\left[f_j[r]\right]$ and $\left[\widetilde{f_j}[r]\right]$ of $\boldsymbol{D}$ by Equation (12) [18]:
    12. $\begin{cases} f_j[r] = \sum_{i=1}^{m\prime} [\![ y_j \in \boldsymbol{Y}_i ]\!] [\![ \delta_j(\boldsymbol{x}_i) = r ]\!] \\ \widetilde{f_j}[r] = \sum_{i=1}^{m\prime} [\![ y_j \notin \boldsymbol{Y}_i ]\!] [\![ \delta_j(\boldsymbol{x}_i) = r ]\!] \end{cases},\ (0 \le r \le k);$ (12)
13. endfor //Here $s$ is a smoothing parameter ($s$ = 1 results in Laplace smoothing); $\delta_j(\boldsymbol{x}_i)$ is the number of $\boldsymbol{x}_i$'s neighbors with positive label $j$.
14. Identify $k$-nearest neighbors $\mathcal{N}(\boldsymbol{x}_0)$ in $\boldsymbol{D}$ according to Step 6;
15. Calculate the identical label statistics $\left\{C_j, j = 1 \sim q\right\}$ for $\boldsymbol{x}_0$ as in Equation (13):
16. $C_j = \sum_{i=1,\ (x,y) \in \mathcal{N}(x_0)}^{k} [\![ y_j \in \boldsymbol{Y} ]\!];$ (13)
//$C_j$ is the number of $\boldsymbol{x}_0$'s neighbors with positive label $j$.
17. for $j$ = 1 to $q$ do
    18. Calculate the likelihood probability $\mathbb{P}\left(C_j \middle| H_j\right)$ and $\mathbb{P}\left(C_j \middle| \overline{H}_j\right)$ of $x_0$ as in Equation (14) [18]:
    19. $\begin{matrix} \mathbb{P}\left(C_j \middle| H_j\right) = \frac{s + f_j[C_j]}{s(k+1) + \sum_{r=0}^{k} f_j[r]} \\ \mathbb{P}\left(C_j \middle| \overline{H}_j\right) = \frac{s + \widetilde{f_j}[C_j]}{s(k+1) + \sum_{r=0}^{k} \widetilde{f_j}[r]} \end{matrix},\ \left(1 \le j \le q,\ 0 \le C_j \le k\right);$ (14)
    20. Predict label $j$ of $\boldsymbol{x}_0$ according to Equation (15):
    21. $y_{0j} = [\![ \frac{\mathbb{P}(H_j)\mathbb{P}(C_j|H_j)}{\mathbb{P}(\overline{H}_j)\mathbb{P}(C_j|\overline{H}_j)} > 1 ]\!];$ (15)
22. endfor
```

4. Experimental Results and Analysis

4.1. Experimental Data Set

The data was acquired in May of 2018 by MCS in the LTE network of Shanghai, the most well-known city of China which located at the east coast of the country. The dataset includes a total of 729,294 samples contributed by 88,594 users. The target URLs of web browsing involved are 10 top websites in China such as Sina, Sohu, Weibo and Taobao.

Since the KQI prediction discussed in this paper is a classification problem, the original numerical labels $\{D_{dns}, D_{tcp}, D_{req}, D_{res}\}$ of the dataset are converted into Boolean ones according to predefined thresholds. In a real network, in most cases the KQIs in the samples are positive, only a very few of them are negative. The label imbalance in general will impact on the prediction seriously. To mitigate this problem, it is better to restrict the ratio of positive and negative samples within 3:1. Therefore, here we employ the boxplot method to determine appropriate thresholds of the labels. Specifically, for each label we take Q3, namely, the upper quartile in the whole dataset as the threshold. Label values higher than the threshold are marked as negative, and vice versa.

The boxplots of the labels are shown in Figure 1. Since the label values for negative samples are always very high, common logarithm is taken for all the samples for ease of illustration.

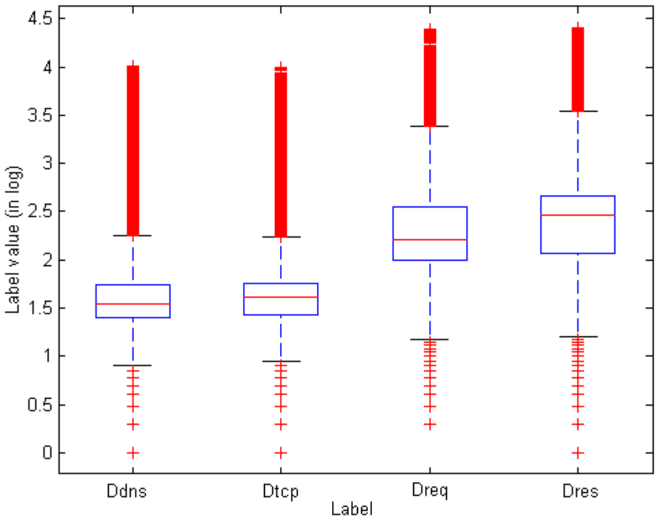


Figure 1. Boxplot of key quality indicators (KQI) labels (unit: ms).

In Figure 2 we show the spatial distribution of the samples (down-sampled for illustration purpose). The samples with null or invalid feature values, or the outliers whose label values are beyond Q3+1.5IQR and Q1−1.5IQR are eliminated. The samples cover almost the whole metropolitan area of about 6340 km^2, including the Chongming Island.

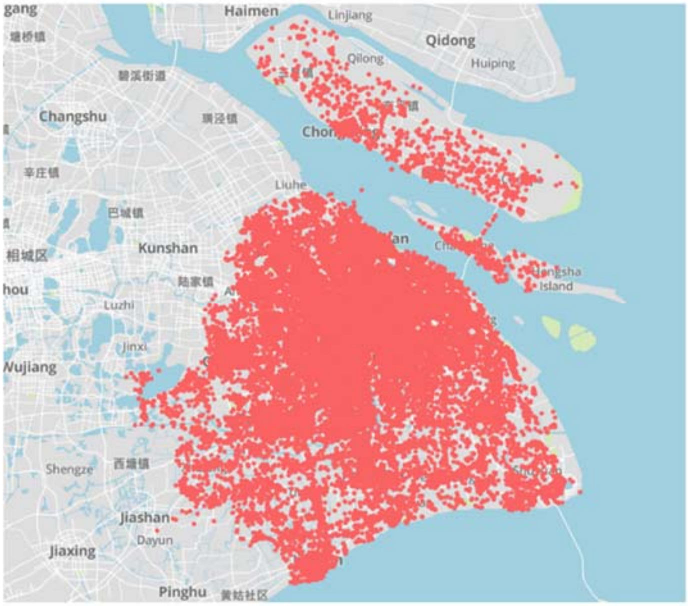


Figure 2. Spatial distribution of data samples.

4.2. *Criteria for Performance Evaluation*

The performance measures of multi-label learning can be divided into two categories, namely, prediction-based and ranking-based indices [23]. The former mainly evaluates the correctness of the label prediction results, including Accuracy, F1-measure, Hamming Loss. The latter evaluates the ranking performance of the labels based on the scoring function f, including One-error, Coverage, Ranking Loss, and Average Precision. In general, prediction-based indicators are more common. Considering that only a few of the labels are involved in the KQI prediction, the ranking-based indices are of little significance for evaluation purpose. Therefore, hereafter only the prediction-based indices are evaluated.

All the experiments are conducted on PyCharm 2.7 platform with Python.

4.3. *Performance of Feature Selection*

First, the ML-ReliefF, RF-ML and UML-ReliefF algorithms are evaluated, and the experimental results are shown in Table 1.

Table 1. Comparison of feature weights of the feature selection (FS) algorithms.

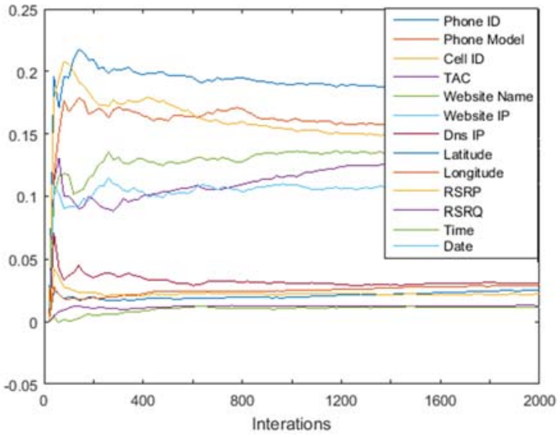
Feature ID	Feature Name	ML-ReliefF	RF-ML	UML-ReliefF
9	Website Name	0.332	0.457	2.593
10	Website IP	0.247	0.802	2.344
3	TAC	0.271	0.473	1.377
13	Phone Model	0.402	1.861	0.734
11	DNS IP	0.065	0.185	0.453
12	Phone ID	0.473	2.491	0.381
4	CellID	0.407	2.484	0.348
6	Latitude	0.062	0.889	0.348
5	Longitude	0.054	1.507	0.283
7	RSRP	0.054	0.765	0.122
1	Date	0.034	0.52	0.119
8	RSRQ	0.029	0.567	0.102
2	Time	0.035	0.359	0.088

In Table 1, the feature weights of UML-ReliefF are listed in the descending order. It can be seen that the weights achieved by different algorithms are significantly different, in both the ranking and weight concentration aspects. By summarizing the weights of Top N features (as shown in Table 2), it is found that the weighting concentration of UML-ReliefF is the highest. For example, the total weight of Top 4 features, i.e., the features with ID = {9, 10, 3, 13}, exceeds 75%.

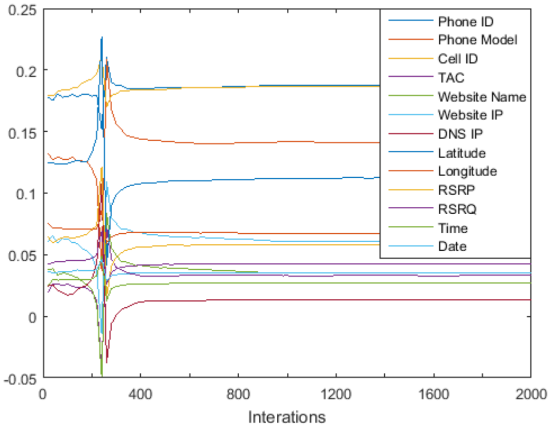
Table 2. Weight concentration of top N features (%).

No. of Features	ML-ReliefF	RF-ML	UML-ReliefF
1	19.19	18.65	27.91
4	65.48	62.45	75.86
7	89.13	80.83	88.59
10	96.03	92.50	96.70
13	100.00	100.00	100.00

In addition, the three FS algorithms also have different convergence rates during the weight iterations (as shown in Figure 3). It can be seen that RF-ML converges after about 500 iterations, which is the fastest, while UML-ReliefF converges at about 1000 iterations, and ML-ReliefF does not converge even after 2000 iterations.

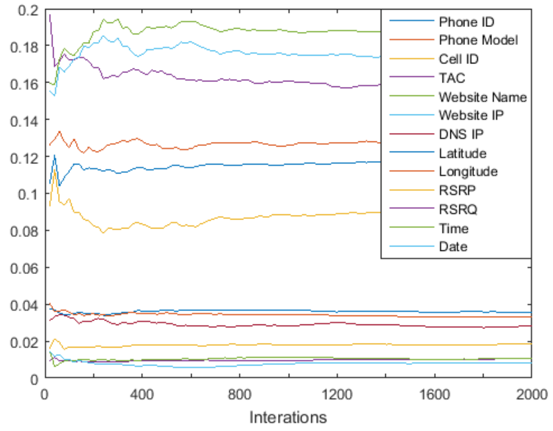


(**a**) ML-ReliefF



(**b**) RF-ML

Figure 3. *Cont.*



(**c**) UML-ReliefF

Figure 3. Convergence of normalized weights for the three FS algorithms.

The impact of feature selection algorithms on the KQI prediction is evaluated and analyzed by the following experiments.

4.4. Experiments of KQI Prediction

In this section, we evaluate the performance of the proposed KQI prediction algorithm. The training set and the test set are segmented by a sliding window (as shown in Figure 4). That is, under the preset prediction window length L (unit: day), the training set and the test set are obtained by segmenting the whole dataset according to feature "Date". The proposed KQI prediction algorithm is then evaluated with the two datasets. The experiment is repeated by sliding the window throughout the whole dataset. Finally, the mean and variance of the evaluation indices can be obtained.

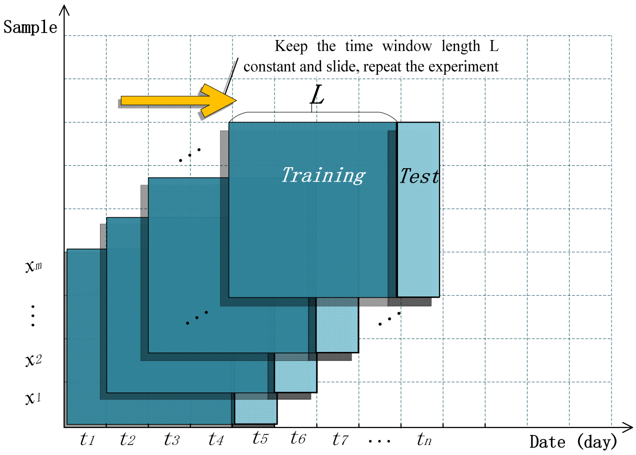

Sample
Keep the time window length L
constant and slide, repeat the experiment
L
Training
Test
xm
x2
x1
t1
t2
t3
t4
t5
t6
t7
. . .
tn
Date (day)

Figure 4. Segmentation of dataset.

(1) KQI prediction with unweighted ML-kNN

First, the performance of KQI prediction with the conventional ML-kNN is investigated. In the experiments, the TOP N (N = 1, 4, 7, 10, 13) features are retained to participate in the KQI prediction. The number of nearest neighbors k in the neighbor searching is taken as 5, 10 and 15 respectively, and k = 15 gives the best result. The experimental results of KQI prediction under the condition of ML-ReliefF, RF-ML, and UML-ReliefF are shown in Table 3.

Table 3. Experimental results of KQI prediction with unweighted ML-kNN (k = 15, L = 14).

No. of Features		Hamming Loss	Macro-F1	Micro-F1	Instance-F1
ML-ReliefF	1	0.299 ± 0.007	0.817 ± 0.006	0.824 ± 0.005	0.800 ± 0.005
	4	0.176 ± 0.004	0.877 ± 0.005	0.881 ± 0.004	0.868 ± 0.003
	7	0.161 ± 0.001	0.887 ± 0.004	0.891 ± 0.002	0.878 ± 0.001
	10	0.164 ± 0.005	0.886 ± 0.005	0.889 ± 0.004	0.875 ± 0.004
	13	0.162 ± 0.003	0.887 ± 0.004	0.890 ± 0.003	0.876 ± 0.002
RF-ML	1	0.299 ± 0.007	0.817 ± 0.006	0.824 ± 0.005	0.800 ± 0.005
	4	0.287 ± 0.004	0.817 ± 0.005	0.825 ± 0.004	0.802 ± 0.004
	7	0.181 ± 0.011	0.874 ± 0.010	0.879 ± 0.008	0.864 ± 0.008
	10	0.183 ± 0.013	0.873 ± 0.012	0.878 ± 0.009	0.863 ± 0.009
	13	0.162 ± 0.003	0.887 ± 0.004	0.890 ± 0.003	0.876 ± 0.002
UML-ReliefF	1	0.187 ± 0.017	0.876 ± 0.006	0.878 ± 0.008	0.863 ± 0.012
	4	0.163 ± 0.004	0.887 ± 0.004	0.890 ± 0.003	0.877 ± 0.003
	7	0.161 ± 0.001	0.887 ± 0.004	0.891 ± 0.002	0.878 ± 0.001
	10	0.164 ± 0.005	0.886 ± 0.005	0.889 ± 0.004	0.875 ± 0.004
	13	0.162 ± 0.003	0.887 ± 0.004	0.890 ± 0.003	0.876 ± 0.002

It can be seen that for UML-ReliefF, TOP 4 features lead to almost the best result, while for ML-ReliefF, TOP 7 features give the best result and for RF-ML all the 13 features should be employed to make the best prediction. In case the prediction with only TOP 1 feature, UML-ReliefF also performs much better than the other two. Therefore, UML-ReliefF is capable of achieving the most reasonable ranking and weights for the features.

The KQI prediction results with TOP 7 to 13 features for UML-ReliefF and ML-ReliefF are exactly the same, because the features selected for both the cases are the same, even though the ranking and weights are different.

(2) KQI prediction with feature-weighted ML-kNN

Next, the experiments of KQI prediction with feature-weighted ML-kNN and UM-ReliefF is carried out. The results are presented in Table 4.

Table 4. Experimental results of KQI prediction with feature-weighted ML-kNN (k = 15, L = 14) and UML-ReliefF.

No. of Features	Hamming Loss	Macro-F1	Micro-F1	Instance-F1
1	0.187 ± 0.017	0.876 ± 0.006	0.878 ± 0.008	0.863 ± 0.012
4	0.166 ± 0.003	0.885 ± 0.004	0.888 ± 0.003	0.875 ± 0.003
7	0.167 ± 0.004	0.883 ± 0.004	0.887 ± 0.003	0.873 ± 0.003
10	0.163 ± 0.005	0.887 ± 0.004	0.890 ± 0.003	0.876 ± 0.004
13	0.159 ± 0.002	0.888 ± 0.004	0.892 ± 0.003	0.878 ± 0.002

Compared with the results of unweighted prediction in Table 4, it is seen that feature-weighted ML-kNN prediction gives more satisfactory and stable results, since all the features contribute to the prediction according to their weights. Therefore, more the number of features that get involved in the learning, the better performance it will be. On the other hand, for traditional filter feature selection, which does not participate in the learning, one needs to select carefully as to how many features are to be retained for the learning. It should be noted that feature-weighted learning is adequate for the case of not too many features. For high-dimensional dataset, we still need to balance the learning performance and the computational complexity.

(3) Influence of window length L on KQI prediction

As the prediction window length is an important parameter in feature-weighted KQI prediction, we need to find its optimal value. Here, we divide the dataset according to different L and compare its influence on the KQI prediction.

According to the experimental results (as shown in Figure 5), except that RF-ML is not sensitive to the window length, the other two FS algorithms reach the best performance with a window length of 14. Therefore, in practice the KQI prediction with L = 14 is suggested.

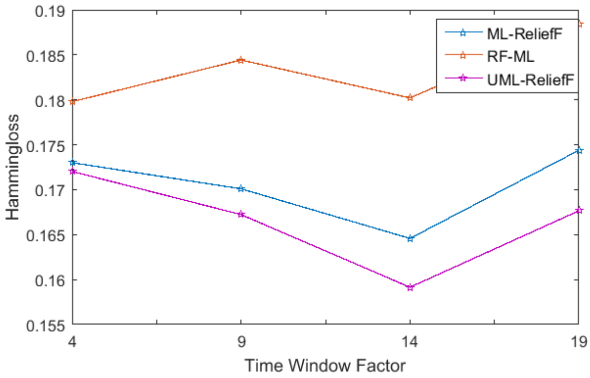


Figure 5. Comparison of KQI prediction performance under different settings of window length.

5. Conclusions

Currently, mobile network maintenance relies heavily on manual investigation and customer feedback. It is incapable of finding service deterioration in a timely manner and, thus, engineers cannot take prompt actions to solve it before it is too late. There is some research of service quality prediction reported in the area of web service recommendation [3,10,11]. Most of them are designed for low dimensional data and assume that the feature and label space is ideally independent. There is limited research of OTT service prediction for high-dimensional real data in mobile networks. In view of this, this paper tries to employ feature selection and multi-label learning in the prediction of OTT service quality, which would be highly valued information for engineers.

Based on the mobile crowdsensing data acquired from massive smartphone users, we have first investigated the application of multi-label ReliefF in the feature selection and proposed an improved ML-ReliefF, namely, UML-ReliefF. Then, a feature-weighted ML-kNN algorithm was proposed for KQI prediction, by combining UML-ReliefF and ML-kNN. Unlike the traditional filter FS, which does not participate in the learning process that follows, here the weights of the selected features are utilized in the prediction. We call it embedded filter feature selection.

Taking OTT web browsing service as an example, experimental results of KQI prediction show that UML-ReliefF is able to overcome the drawback of ML-ReliefF, which does not consider the relations among the labels. Second, it is also proved that feature-weighted KQI prediction, which takes full advantage of all the features, yields better performance than its unweighted counterpart. This is especially meaningful for the case, where the dimension of the features is not so high and computational complexity is tolerable.

The proposed method of this paper provides a promising way for the mobile network operation engineers to tackle service deterioration as early as possible, and thus may improve the intelligence of mobile network operations.

In the future, we will further extend the KQI prediction into regression problem. That is, not just predicting whether the service quality will be “good” or “bad”, but finding the actual KQI values instead. Another challenge that deserves further investigation is as to how to utilize the soft information embedded in the original KQI labels before hard decision. It is likely to improve the KQI prediction, if we can take advantage of it in a proper way. In addition, the impact of distance measures on the performance of nearest neighbour searching in the KQI prediction will also be investigated.

Author Contributions: Conceptualization, K.L. and W.L.; methodology, K.L. and S.X.; validation, S.X. and M.X.; visualization, S.X.; writing, K.L. and S.X.; data curation, K.L. and W.L.; funding acquisition, M.X. and K.L.; writing-review & editing, L.Z., K.L. and S.X. All authors have read and agree to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (61841601, 61972040), the Collaborative Innovation Project of Chaoyang District of Beijing (CYXC1805), the Premium Funding Project for Academic Human Resources Development in Beijing Union University (BPHR2018CZ05), and Beijing Municipal Education Commission Science and Technology Project (KM201911417010).

Acknowledgments: The authors would like to thank M.N.S. Swamy of Concordia University, Canada, for going through the paper and making a number of suggestions, which has helped in improving the presentation of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, S.; Lin, Y.; Li, S.; Chen, Z.; Wan, H. Deep Spatial–Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3913–3926. [CrossRef]
2. Zhang, C.; Patras, P. Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks. In Proceedings of the 19th MobiHoc, Los Angeles, CA, USA, 26–29 June 2018; pp. 231–240.
3. Wang, S.; Ma, Y.; Cheng, B.; Yang, F.; Chang, R.N. Multi-Dimensional QoS Prediction for Service Recommendations. *IEEE Trans. Serv. Comput.* **2019**, *12*, 47–57. [CrossRef]
4. Ganti, R.; Ye, F.; Lei, H. Mobile crowdsensing: Current state and future challenges. *IEEE Commun. Mag.* **2011**, *49*, 32–39. [CrossRef]
5. Siegfried, G. Beginning AIOps: Data Science for IT Operations. Available online: https://www.gartner.com/doc/3893177?ref=mrktg-srch (accessed on 5 February 2019).
6. Casas, P.; Seufert, M.; Wamser, F.; Gardlo, B.; Sackl, A.; Schatz, R. Next to You: Monitoring Quality of Experience in Cellular Networks from the End-Devices. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 181–196. [CrossRef]
7. Li, K.; Wang, H.; Xu, X.; Du, Y.; Liu, Y.; Ahmad, M.O. A Crowdsensing Based Analytical Framework for Perceptional Degradation of OTT Web Browsing. *Sensors* **2018**, *18*, 1566. [CrossRef] [PubMed]
8. Bulut, M.; Demirbas, M.; Ferhatosmanoglu, H. LineKing: Coffee Shop Wait-Time Monitoring Using Smartphones. *IEEE Trans. Mob. Comput.* **2015**, *14*, 2045–2058. [CrossRef]
9. China Telecom. *Functional specification of OTT service perception test APP of China Telecom*; China Telecom: Beijing, China, 2015.
10. Zheng, Z.; Ma, H.; Lyu, M.; King, I. QoS-Aware Web Service Recommendation by Collaborative Filtering. *IEEE Trans. Serv. Comput.* **2010**, *4*, 140–152. [CrossRef]
11. Tang, M.; Jiang, Y.; Liu, J.; Liu, X. Location-aware collaborative filtering for QoS-based service recommendation. In Proceedings of the 19th International Conference Web Services (ICWS'12), Honolulu, HI, USA, 24–29 June 2012; pp. 202–209.
12. Kononenko, I. Estimating attributes: Analysis and extension of relief. In Proceedings of the European Conference on Machine Learning, Berlin/Heidelberg, Germany, 6–8 April 1994; pp. 171–182.
13. Wang, Z.; Zhang, Y.; Chen, Z.; Yang, H.; Sun, Y.; Kang, J.; Yang, Y.; Liang, X. Application of ReliefF algorithm to selecting feature sets for classification of high resolution remote sensing image. In Proceedings of the 2016 IEEE Int'l Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 755–758.
14. Huang, Z.; Yang, C.; Zhou, X.; Huang, T. A hybrid feature selection method based on binary state transition algorithm and ReliefF. *IEEE J. Biomed. Health Inform.* **2018**, *23*, 1888–1898. [CrossRef] [PubMed]
15. Huang, L.-L.; Tang, J.; Sun, D.; Luo, B. Feature selection algorithm based on multi-label ReliefF. *J. Comput. Appl.* **2012**, *32*, 2888–2890. [CrossRef]
16. Spolaôr, N.; Cherman, E.A.; Monard, M.; Lee, H.D. ReliefF for Multi-label Feature Selection. In Proceedings of the 2013 IEEE Brazilian Conference on Intelligent Systems, Fortaleza, Brazil, 19–24 October 2013; pp. 6–11.

17. Zhang, M.; Zhou, Z. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognit.* **2007**, *40*, 2038–2048. [CrossRef]
18. Zhang, M.; Zhou, Z. A Review on Multi-Label Learning Algorithms. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1819–1837. [CrossRef]
19. Li, F.; Miao, D.; Zhang, Z.; Zhang, W. Mutual information based granular feature weighted k-Nearest Neighbors algorithm for multi-label learning. *J. Comput. Res. Dev.* **2017**, *54*, 1024–1035.
20. He, X. *Multivariate Statistical Analysis*, 5th ed.; China Renmin University Press: Beijing, China, 2019.
21. Locarek-Junge, H.; Weihs, C. Classification as a Tool for Research. In Proceedings of the 11th IFCS (International Federation Of Classification Societies) Biennial Conference and 33rd Annual Conference of the Gesellschaft für Klassifikation e.V., Dresden, Germany, 13–18 March 2009.
22. Saini, I.; Singh, D.; Khosla, A. QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases. *J. Adv. Res.* **2013**, *4*, 331–344. [CrossRef] [PubMed]
23. Wu, X.; Zhou, Z. A Unified View of Multi-Label Performance Measures. In Proceedings of the 34th Int'l Conference on Machine Learning (ICML'17), Sydney, Australia, 6–11 August 2017; pp. 3780–3788.