International Journal of Innovative
Computing, Information and Control
Volume **14**, Number **6**, December **2018**

 ISSN 1349-4198
pp. 2175–2188

F-MEASURE: A FORECASTING-LED TIME SERIES DISTANCE MEASURE IN LARGE-SCALE FORECASTING OF VIDEO SERVICES PERFORMANCE

Yu Zhuo[1,2], Jiali You[1], Hanxing Xue[1,2] and Jinlin Wang[1]

[1]National Network New Media Engineering Research Center
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
{ zhuoy; youjl; xuehx; wangjl }@dsp.ac.cn

[2]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

Received January 2018; revised May 2018

Abstract. *To improve the quality of online video services, a measurement and recommendation system for online video services (MCS) was previously developed by the authors, which monitors resource performance for different users of different websites. In the traditional quality of service (QoS) evaluation method, performance is measured separately for each video, generating excessive time and bandwidth costs. Therefore, a measurement approach based on cluster analysis has been proposed by the authors. In this paper, a forecasting-led time series distance measure used in the clustering process of this measurement approach is proposed, named F-measure. This measure is adapted to the features of the measurement results. According to experiment, the error coefficient of this measure is smaller than that for the Euclidean distance. Further, this approach reduces the workload of a large-scale measurement to the 1/Lth original scale at least in active measurement, when the cluster level is L. Thus, it becomes possible for large-scale continuous online video measurement to generate low interference.*
Keywords: Online video, QoS, Time series cluster, F-measure, Large scale measurement

1. **Introduction.** In recent years, large-scale video services are responsible for the largest portion of Internet traffic. For example, Google has been reported to be the largest source of Internet traffic in the world, and YouTube video delivery is an important part of Google's growing traffic [1]. In China, there are 579 million online video users by the end of 2017, which counts 75% of all Internet users [2]. With the rapid expansion of online video services, the providers are springing up like mushroom. Studies for their QoS are got more attention. Different from traditional Internet services, quality evaluations of online video services consider not only delay issues, but also bandwidth and re-buffering [3,4].

Video websites usually provide a large amount of contents. Measurement of every video for every user in round-robin costs so much time and bandwidth. Measuring videos by several groups is a familiar way to reduce the measurement cost. Most of existing works about large-scale video measurement regarded the contents, which are provided by the same website or distributed by the same CDN, as the same. In these works, they measured websites [5,6] or their CDNs [7-9]. Classifying videos by their metadata (the attributes

DOI: 10.24507/ijicic.14.06.2175

of online video services) [4] is also a common way. However, the QoS of exact content cannot be measured well by these methods.

In long-term QoS-aware cloud service composition, time series data mining has been introduced. QoS of cloud services are represented by time series mostly [10]. With this method, some service composition framework is proposed to select the optimal compositions of cloud services for end users [11].

Different from that, this work introduces time series data mining to online video QoS study. URLs are clustered (hereinafter, "URL" refers to specific online video service content, as represented by a play page URL) into a few classes by quality performance measurement results. As discussed in prior work [12], online video websites provide content through content delivery networks (CDNs). Because the number of CDNs and their scheduling strategies are limited, few different service performance variation patterns are expected; therefore, these URLs can be clustered into several service classes. In each class, the URLs exhibit similar performance.

Quality changes in video communication and broadcasting are not completely random. Therefore, the online video QoS is partly predictable. Thus, one series can be predicted based on another, and this can be used to measure the distance between two time series of online video service quality.

In a previous study [13], the authors proposed a large-scale measurement approach based on video clustering, referred to as the measurement and recommendation system for online video services (MCS). In this method, the service performance for each URL is measured several times, beginning immediately after it has been provided by the website. Then, URLs are clustered based on their measured performance. After clustering, URLs in the same class are regarded as providing the same QoS and treated as one URL.

In this paper, in-depth development of this measurement approach is reported, with a forecasting-led time series distance measurement algorithm named F-measure being proposed. In this measure, the similarity (or distance) between two time series is defined, based on the error of one prediction relative to the other, as determined by an exact forecasting algorithm. This measure closely combines time series clustering and forecasting, which can be adapted to the features of online video service QoS data very well. This measure adapts to the data characteristic of the online video quality. The URLs are clustered using a hierarchical clustering method and the developed measure in the measurement approach. Then, the error of this algorithm is discussed. The overall forecasting error (OFE) is also computed to study whether this measure adapts to its usage scenario. Finally, different measures are compared based on their OFEs. The measure proposed in this paper requires fewer data items than a typical measure, for equivalent OFEs.

The main contribution of this work is to propose a forecasting-led time series distance measurement algorithm. This algorithm has the following characteristics: 1) with this algorithm using in our measurement approach, the measurement-scale reduction yields a smaller error; 2) it is highly adapted to the features of online video service QoS data; 3) a clustering-based measurement approach incorporating this measure is designed to measure the performance of each service more individually, allowing service performance for particular content delivery to be predicted with greater accuracy. In addition, this concept can be expanded for application to quality measurement for other services with many similarities.

The remainder of the article is structured as follows. In Section 2, the MCS designed and realized by the authors is introduced. The MCS results are analyzed and certain features of these data are summarized. Section 3 summarizes previous research on large-scale measurement of online video service QoS and existing time series similarity measures, with analysis of their unfeasibility for application in this work. Section 4 presents the design

of the time series similarity measure proposed in this paper, while Section 5 describes the experimental design employed in this study and reports on a comparison of the developed measure with existing methods. Finally, Section 6 concludes the article and discusses future work.

2. Measurement System and Data.

In this section, an overview of the MCS is provided and the video metadata are presented. Then, the data features are discussed.

2.1. Measurement system.

In prior work [13], the MCS are designed, which measures online video QoS and collects the results for further analysis. The structure of MCS is shown in Figure 1. The system is divided into the following three components.

Metadata crawling subsystem: This subsystem primarily collects video-service URLs and meta data. The URLs collected by this subsystem are used in the next step, QoS forecasting, after clean up. The metadata used here is video resolution, and the resolutions are divided into four classes: 1) Ultra high definition (HD) (1920 × 800-1920 × 1072), 2) HD (1280 × 536-1280 × 720), 3) Standard definition (SD) (640 × 272-1024 × 432), and 4) Smooth (480 × 208-640 × 272).

QoS parameter measurement subsystem: This subsystem is used for URL measurement. It then plays a role in data collection, clean up, and reservation. A large number of measurement nodes provide quality metrics results of different locations and different times. These nodes are located at the edge of the network, including traditional or mobile terminal, intelligent routing, smart TV and so on.

Service recommendation subsystem: This subsystem recommends service sources to users. When a user selects a video to access, this subsystem gathers QoS forecasting results from different online video websites and recommends the website with the best video QoS to the user.

In this measurement system, a method that simulates user access behaviors is used. First, the measurement node (e.g., measurement server or user terminal) requests a video

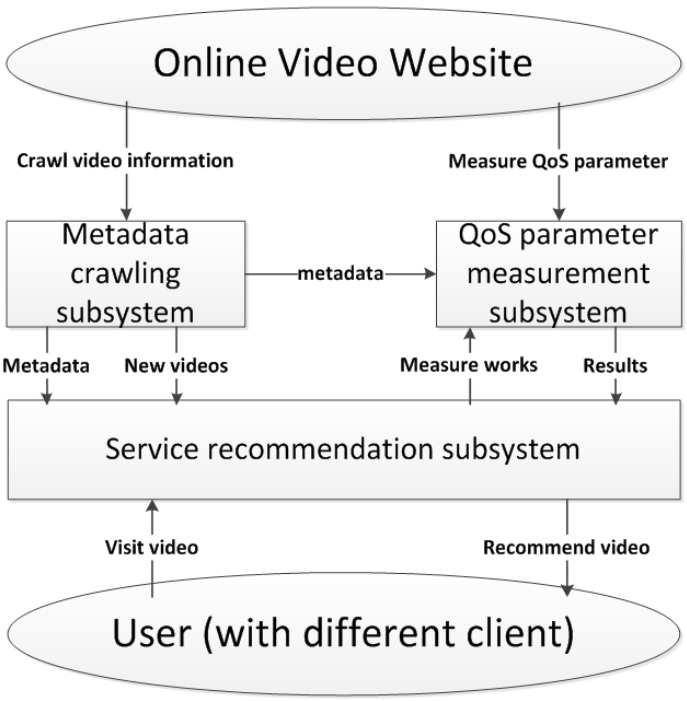

FIGURE 1. MCS structure

segment to "watch". After this request, the node records the delay and download time. Then, the node calculates the other quality parameters from the measurements conducted during downloading. In this study, to establish a balance between reducing the measurement error and improving efficiency, the download size from each video was set to 50 KB. For a common download rate is about 100-10000 KB/s, download 50 KB needs less than 1 s, which can reflect the momentary rate of download. The random error will raise as the download size decreases; and the larger download size will lead to more time for measurement relatively, causing the results to reflect more about the average rate other than the momentary rate. It was pointed out that 1 s delay might reduce 6% users. Most users gave up watching when setup delay is more than 2 s [4]. So the timeout period is set 10 s. Most of users cannot wait longer. Further, the delay and download rate for each resolution type and for each video service were used. Under the above parameters, a period of 1-4 s on average was required to measure each video in a single thread on a single node.

2.2. **Measured data features.** The features of the measured data collected by the MCS are analyzed before further study.

Sparsely: For each URL, the results are sparse. If all URLs are measured in a round-robin, the more URLs there are, the longer the distance between two results of the same URL is.

In a large-scale measurement process, the measurement costs are very high, as a large number of videos are crawled; in this study, 1.84 million effective content items were collected from 14 different online video websites. To achieve effective recommendations, predictions of sufficient accuracy must be achieved, which means accurate measurement results must first be obtained. In order to ensure measurement accuracy, multiple measurements must be performed for each URL on multiple measurement nodes. In addition, measurements should also be performed for the URL at a certain measurement density. Here, the measurement density is the reciprocal of the least time interval between each measurement pair. This yields a very large measurement scale.

Considering the time and bandwidth costs discussed above, in this study, the URLs of one website were measured at a maximum rate of 0.5 s/URL on one measurement node. The security settings of online video websites were also considered. Here, 569,856 URLs from Youku, the most popular video site and the main data source in this study, were crawled. Under this measurement speed, the Youku URLs could be measured once every 2 weeks, i.e., once in each complete round. If the download size for each resolution and each URL was 50 kB, the average bandwidth occupied by this Youku measurement was 188 kB/s. However, the highest achievable measurement density was far less than this value for many reasons, such as request timeouts, poor network conditions, and URL failure. Therefore, when all videos were measured, the measurement results for a single video were extremely sparse.

Variable Time Interval: For the same URL and same definition, the interval between adjacent results is variable. The length of this interval is influenced by many different factors. When a video is newly accessible or very popular, this URL may be measured more frequently, and vice versa. Further, because of network variance, effective results cannot be obtained for some measurements. These data are removed in the data cleaning process.

Time Sensitivity: An online video service has busy and idle periods. Suppose that there are two URLs, each exhibiting superior performance during busy and idle periods, respectively. It is apparent that these services will exhibit very different performance levels at most times.

In some scenarios, for example, those involving speech recognition, researchers require similarity measures insensitive to time. This is because different people may utter the same word in different ways and at different speeds. Many measures have been developed for such scenarios, such as dynamic time warping (DTW) distance. However, for investigation of online video service performance, a time sensitive measure is required.

Different Timestamps: Obviously, every URL cannot be measured at the same time and place. Therefore, different results have different timestamps. Thus, the time series of different URLs or different definitions cannot be closely aligned based on their timestamps.

Existing methods cannot be applied to data with these features, as discussed in the next section.

3. **Existing Algorithms.** Time series clustering is a fundamental branch of time series data mining. Although many pattern recognition or data analysis methods employ time series clustering, most of these techniques are applied in the fields of finance, medicine [14], energy [15-17], network [18] and so on. In this paper, this approach is applied to the field of online video service QoS measurement. Time series clustering in this field can divide the URL with their quality performance characteristic. This means the cost of large-scale measurement will be reduced significantly, and the additional network pressure caused by measurement will be decreased as well.

Time series clustering algorithms can be divided into three categories [19,20] as outlined in Figure 2. The function used to measure the similarity between two compared time series is a key component of clustering. These data can be in various forms, including raw values of equal or unequal length, vectors of feature-value pairs, and transition matrices. In this paper, each dataset forms a time series combined with real numbers.

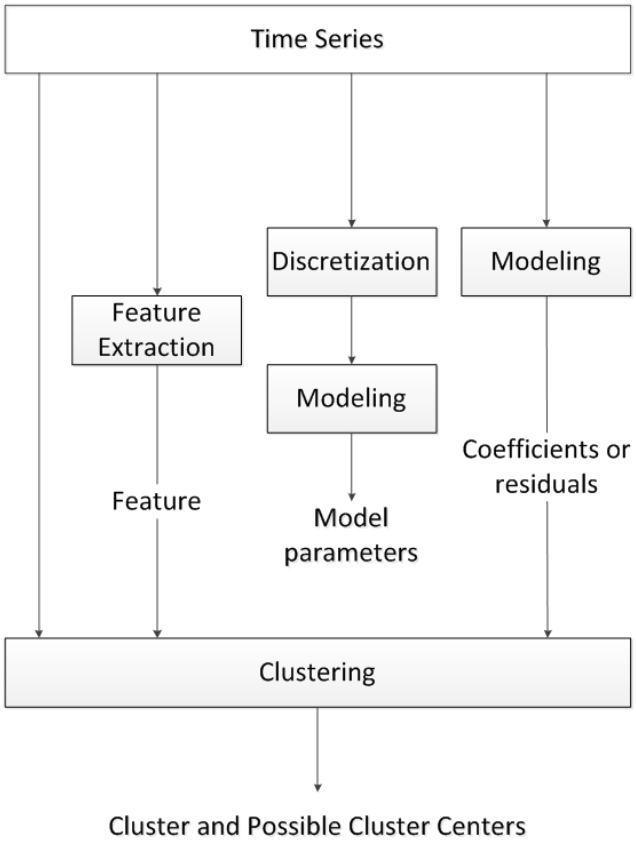

FIGURE 2. Three time series clustering approaches

Measures having strong similarities are generally classified into four categories: shape-, edit-, feature-, and structure-based measures, as discussed below.

Shape-based measures calculate the distance between two series based on specific points of the time series. Typical algorithms include the Euclidean distance (Euclidean) [21], DTW [22,23], and approximate shape exchange algorithm (ASEAL) [24].

An edit-based measure is used to calculate minimum operations when one time series is changed to another. These operations are generally "insert", "change", "delete", and so on. Edit distance with renal penalty (ERP) is an example of a typical algorithm of this kind [25].

A feature-based measure calculates distance based on important characteristic values of the time series. The bag-of-patterns [26] and complexity invariant distance (CID) [27] algorithms are examples, and many more measures of this kind exist. Some measure combined few different measures to adapt both short and long time series clustering [28].

Finally, a structure-based measure finds the structures of series at large distances and compares them in the context of the overall situation. A typical example is based on the autoregressive moving average (ARMA) model [29].

In Table 1, significant features of similar measures are summarized and compared against the required features discussed in Section 2. In this table, each row represents one typical measure and each column represents a feature mentioned in Section 2. Further, "Y" and "N" respectively indicate that the given measure does or does not support this feature. It is apparent from Table 1 that the existing time series distance measures cannot be used in this large-scale measurement approach. Therefore, a new forecasting-led time series distance measure for this scenario is proposed in this paper.

Table 1. Significant measures and their features

	Sparse	**Variable Interval**	**Sensitive to Time**	**Different Time**
Euclidean	Y	Y	Y	N
DTW	Y	Y	N	Y
ASEAL	Y	Y	N	Y
ERP	Y	Y	N	N
Bag-of-Patterns	N	Y	N	Y
CID	Y	Y	Y	N
ARMA	Y	N	Y	N
F-measure	Y	Y	Y	Y

4. **Forecasting-Led Time Series Clustering Approach.** As discussed in the previous section, a new time series distance measure for online video service performance measurement is proposed in this study. In this section, the distance measure is described and its features are discussed. Sections 4.1 and 4.2 give a definition of this measure and present some of its features, respectively.

4.1. **Forecasting-led time series distance measure.** In this subsection, the measure proposed in this paper is introduced.

A group G means a dataset of the time series that will be clustered. $G = \{T_1, T_2, \ldots, T_N\}$, where N is the number of time series in G. Any $T_i = \{(x_{i1}, t_{i1}), (x_{i2}, t_{i2}), \ldots, (x_{iM_i}, t_{iM_i})\}$ is a time series, where the time variable t_{ia} is the time at which the value of the main variable x_{ia} is obtained. M_i is the length of T_i.

A basic forecasting algorithm, which we refer to using the placeholder "Fpre", is needed in this forecasting-led measure. Here, $y_k = Fpre(t_k, T_i)$ is the forecast main variable value at time t_k, which is predicted by Fpre using T_i.

Definition 4.1. *For any time series T_i and T_j, the "+" operator is defined as*

$$T_i + T_j = \left\{\left(x_{(i+j)1}, t_{(i+j)1}\right), \left(x_{(i+j)2}, t_{(i+j)2}\right), \ldots, \left(x_{(i+j)M_{i+j}}, t_{(i+j)M_{i+j}}\right)\right\}, \tag{1}$$

where $\{(i+j)1, (i+j)2, \ldots, (i+j)M_{i+j}\} = \{i1, i2, \ldots, iM_i\} \cup \{j1, j2, \ldots, jM_j\}$ *and*

$$x_{(i+j)k} = \begin{cases} x_{ik_i}, & when\ (i+j)k \in \{i1, i2, \ldots, iM_i\},\ ik_i = (i+j)k, \\ x_{jk_j}, & when\ (i+j)k \in \{j1, j2, \ldots, jM_j\},\ jk_j = (i+j)k, \\ \dfrac{x_{ik_i} + x_{jk_j}}{2}, & when\ (i+j)k \in \{i1, i2, \ldots, iM_i\} \cap \{j1, j2, \ldots, jM_j\}, \\ & ik_i = jk_j = (i+j)k. \end{cases} \tag{2}$$

Definition 4.2. *For time series T_i and data $\{x_{ik}, t_{ik}\}$, the "−" operator is defined as*

$$T_i - \{(x_{ik}, t_{ik})\} = \{(x_{i1}, t_{i1}), \ldots, (x_{ik-1}, t_{ik-1}), (x_{ik+1}, t_{ik+1}), \ldots, (x_{iM_i}, t_{iM_i})\}. \tag{3}$$

Definition 4.3. *For any time series T_i and T_j, the following measure is defined:*

$$D(T_j, T_i) = \frac{\sum_{l=N_{Fpre}+1}^{|T_i+T_j|} \left| x_{(i+j)l} - Fpre\left(t_{(i+j)l}, T_{B(i+j)l}\right) \right|}{|T_i + T_j| - N_{Fpre}}, \tag{4}$$

where $T_{B(i+j)l} = \begin{cases} \left(T_i - \left\{\left(x_{(i+j)l}, t_{(i+j)l}\right)\right\}\right) + T_j, & when\ \left\{\left(x_{(i+j)l}, t_{(i+j)l}\right)\right\} \in T_i, \\ \left(T_j - \left\{\left(x_{(i+j)l}, t_{(i+j)l}\right)\right\}\right) + T_i, & when\ \left\{\left(x_{(i+j)l}, t_{(i+j)l}\right)\right\} \in T_j \end{cases}$, $\forall\{(x_{ml}, t_{ml})\} \in T_{B(i+j)l}$, $t_{ml} < t_{(i+j)l}$. *Here, N_{Fpre} is the minimum number of data elements needed for forecasting the next data elements using Fpre. This forecasting-led time series distance measure is hereafter referred to as "F-measure".*

By Definition 4.3, F-measure has the following properties.

Property 4.1.

$$D(T_i, T_i) = 0. \tag{5}$$

Property 4.2. *For any Fpre, there exists*

$$D(T_j, T_i) = D(T_i, T_j). \tag{6}$$

Property 4.3. *When $T_{ik} = T_{jk}$, for any $k \le M_i$, $M_i = M_j$, there exists*

$$D(T_j, T_i) = \frac{\sum_{l=1}^{M_j} L\left(x_{jl} - x_{il}\right)}{|T_j|} = \frac{\sum_{l=1}^{M_j} \left(x_{jl} - x_{il}\right)}{|T_j|}, \tag{7}$$

which means the F-measure degenerates to the Euclidean distance at this point.

Figure 3 shows the F-measure obtained using the Fpre algorithm with $N_{Fpre} = 5$. In this figure, $dx_{ij} = |x_{ij} - Fpre(t_{ij}, (T_1+T_2)_{Bij})|$, $D(T_1, T_2) = (dx_{23}+dx_{24}+dx_{14}+dx_{25}+dx_{15})/5$.

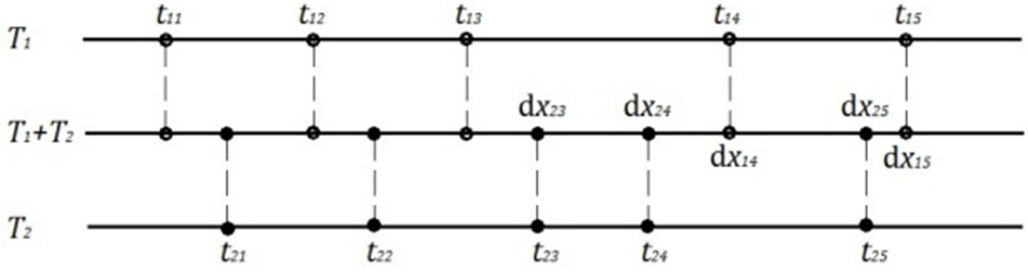

FIGURE 3. Sketch map for F-measure with $N_{Fpre} = 5$

4.2. **F-measure features.** In this subsection, the features of F-measure are discussed, in the context of the four data features mentioned in Section 2.

4.2.1. *Sparse data.* F-measure works well when the time series data are sparse. When $|T_i|$ and $|T_j|$ are small, this measure can give a reasonably accurate distance for T_i and T_j.

In fact, the adaptability for sparse initial data is inherited from the basic forecasting algorithm, i.e., Fpre. If Fpre requires a small dataset and low-density data to provide a forecast, F-measure, which is based on this forecasting algorithm, can then measure the time series for a small dataset and long data time interval.

4.2.2. *Different time intervals of time series.* With the measurement approach used in this paper, a typical measurement result series for a given URL can be represented as follows:

$$\begin{aligned} T_i = \big\{ &\big(x_{i1}, t_{i1}\big), \big(x_{i2}, t_{i1} + \Delta t_1\big), \ldots, \big(x_{ia+1}, t_{i1} + a\Delta t_1\big), \big(x_{ia+2}, t_{i1} + a\Delta t_1 + \Delta t_2\big), \\ &\ldots, \big(x_{i(1+a+b+\cdots+c)}, t_{i1} + a\Delta t_1 + b\Delta t_2 + \cdots + c\Delta t_m\big)\big\}, \\ &\forall \Delta t_k,\ k \neq 1,\ \Delta t_1 < \Delta t_k. \end{aligned} \tag{8}$$

This means that, at the initial measurement time (when the URL is newly entered), the time interval between two neighboring measurement results is always shorter than the interval of the following daily measurement. Further, this interval may change in different periods of daily measurement.

For F-measure, if Fpre can perform a forecast with data having different time gaps, F-measure can also provide this prediction.

4.2.3. *Time displacement.* As discussed in Section 2, in this measurement scenario, the measure must be sensitive to time displacements.

Property 4.4. *For any Fpre, there exists an l such that, when $L(Fpre(t_{jl}, T_i)) \neq L(Fpre(t_{kl}, T_i))$, $D(T_k, T_i) = D(T_j, T_i)$ is not established. Further, where $T_k = \{(x_{k1}, t_{k1}), (x_{k2}, t_{k2}), \ldots, (x_{kM_k}, t_{kM_k})\}$, for any l, $t_{kl} = t_{jl} + \Delta t$.*

When Property 4.4 is established, this measure is sensitive to time displacements.

4.2.4. *Different timestamps in two time series.* For the F-measure based on the *Fpre* forecasting algorithm, it is not necessary for the two time series to have the same timestamps.

It is apparent from Definition 4.3 that, regardless of whether T_j, T_i have the same timestamps, the F-measure distance between these two time series can be calculated.

4.3. **Differences from existing algorithms-measure features.** In this subsection, the differences between existing significant measures and F-measure are compared.

Table 1 shows the features of F-measure and some significant distance measures of time series. In this scene, as discussed in Subsection 2.2, a time sensitive measure is required. This feature is the most important. In Table 1, Euclidean, CID, ARMA and F-measure are sensitive to time, too. However, ARMA is a typical structure-based measure, which has a higher request to the structure of the data. It cannot adapt to the time series of variable intervals well. In the end, Euclidean and CID measure calls the measured time series have the same timestamps, which needs a preprocessing to unify timestamps. That step may introduce extra errors. Instead, F-measure can be applied on time series with different timestamps directly.

5. **Experiments.** In this section, the experimental design of this paper is first introduced. Then, the datasets measured by the MCS are considered to compare the F-measure performance with similar existing time series measures.

5.1. **Dataset.** The dataset used in this work contained 74,594 measurement results measured in one measurement node from Aug 1, 2015 to Aug 31, 2015. Fifty-two URLs were measured in this set, being obtained through stratified sampling from the Youku website that incorporated 155 different URLs, all with different resolutions. For each URL, there were 2-4 different accessible resolutions; hereinafter, this property is referred to as the "URL-resolution".

All data were cleaned and incomplete measurement results were first removed. Results outside three standard deviations on either side of the mean were then removed. For the measured QoS parameter delay and download rate, the result value ranges after cleaning were [0 s, 18.64 s] delay and [0 kbps, 40,640 kbps] download rate.

The data measured between Aug 1 and 7 were regarded as the initial data for clustering. Then, data were sampled from the measurement results from Aug 8 to 31; these sample data were regarded as the measurement data when the user visits the video service and used for comparison with the forecast data of the corresponding time.

A fixed number of measurement results were extracted from the test dataset and used to simulate the actual measurement results. For similarity to the sparse measurement results encountered in actual measurement, 100, 250, 400, 550, 700, and 850 measurement results were extracted as the measurement dataset. For the sampling, it was assumed that the user access process followed the Poisson process. Hence, the measurement results were extracted from the measurement dataset to simulate user access.

For each URL-resolution, the measurement results for the initial and sample datasets were arranged by measurement time and organized into a time series. These time series were used in the subsequent experiments. Figure 4 shows some sample time series. Based on these data, the total measurement errors were calculated and compared.

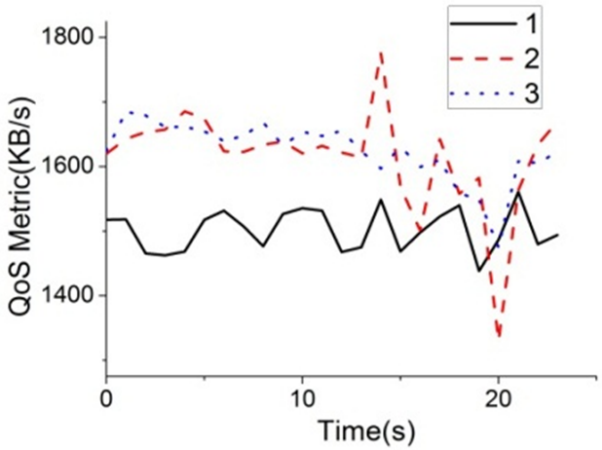

FIGURE 4. Three time series for different URL-resolutions

5.2. **F-measure.** In this study, two common forecasting algorithms were selected as the Fpre algorithm, namely, linear regression and the decision tree method.

5.2.1. *Linear regression.* The forecast model for linear regression is expressed as

$$Y(t) = kt + b + \varepsilon, \tag{9}$$

where k and b are unknown parameters and ε is the stochastic error.

In this study, the ordinary least squares technique (OLS) was used to obtain k and b, with

$$Y(t + \Delta t) - Y(t) = k\Delta t. \tag{10}$$

Therefore, when $k \neq 0$, this model was sensitive to time displacements. As apparent from (9), this algorithm can be used in variance time intervals.

5.2.2. *Decision tree.* The decision tree is a classifier having a tree-like structure. The nodes in this structure are either a leaf (indicating a class), or a decision node that specifies some test to be conducted on a single attribute value, with one branch and sub tree for each possible outcome of the test. A key aspect of the decision tree structure is selection of the attribute and value on the decision node. The choice criterion is a heuristic used to divide the training dataset with a class marker. It determines the topology of the tree and, finally, the choice of split point. In this study, the C4.5 algorithm [30] was used to build the decision tree.

When forecasting with the decision tree method, the data should first be discretized. In online video service performance prediction, the prediction is ultimately used to recommend a video service. As a consequence, even if the forecast results are inaccurate, they meet the requirements in this context if they are sufficiently accurate to allow comparison of different video services. Therefore, reasonable discretization does not affect the application of the forecasting system.

For measurement data that can be observed in cyclic variations, the temporal aspect is discretized to the location of a cycle. Other continuous attributes are discretized to 10 intervals.

5.3. **Evaluation.** In this work, a cluster process was used for prediction of the service QoS. Therefore, the prediction effect of the cluster algorithm was evaluated. That is, the prediction effects of different clustering levels were compared based on their error coefficients. The effects of different clustering levels on the measurement efficiency and accuracy were analyzed.

The error coefficient of a QoS parameter was defined as

$$\alpha = \frac{\sum_{i=1}^{N} (x_i - x_{i0})^2 / (N-1)}{\sum_{i=1}^{N} (x_i - \bar{x})^2 / (N-1)}. \tag{11}$$

In the above formula, N is the total number of forecast results, x_i is the forecast result for a given QoS parameter, and x_{i0} is the measurement result for the same QoS parameter at the same time. Further, $\bar{x}$ is the average value of all measurement results. Therefore, α is the ratio of the mean squared error of each result to the variance of all measurements. As apparent from the definition, when forecasting any results as $x_i = \bar{x}$, $\alpha = 1$.

When the URLs are clustered to several classes and measured for some time, α is expected to be lower than that when the URLs are measured a greater number of times, without clustering.

5.4. **Clustering method.** In this work, a hierarchical clustering method was employed, based on F-measure and the Euclidean distance. As discussed in Subsection 4.3, Euclidean distance can be used to measure the online video performance metrics series with one more preprocessing step. What is more, Euclidean distance is classic and laconic. This measure is used widely in time series cluster. As a consequence, the Euclidean measure is chosen to compare with F-measure.

The forecast approach using Fpre and F-measure was implemented, which can forecast QoS parameters based on the most recent measurement result before the forecast time.

Using this approach, the feasibility of the cluster-based measurement approach could be examined and the conclusions of the above error analysis could be verified.

The specific clustering process is as follows.

(1) Clustering: The URLs are clustered to the appropriate classes based on the initial measurement results. The number of classes defines the cluster level. In this experiment, predictions for four different cluster levels were examined, i.e., 2, 3, 5, and 8. In this way, the cluster-based measurement approach was simulated for large-scale measurement.

(2) Prediction: With any "user access", the forecasting algorithm is applied to forecasting the QoS parameters at different clustering levels.

(3) Error analysis: The forecast error is calculated based on the forecast and measurement results. The prediction error coefficients of all measured results are calculated.

Figure 5 shows a comparison of the error coefficients for the Euclidean distance and F-measure for the two Fpre algorithms mentioned in Subsection 5.2 (linear regression and decision tree). It is apparent from Figure 5 that, regardless of whether linear regression or the decision tree is used as the basic forecasting algorithm, F-measure yields smaller error coefficients in this experiment.

Furthermore, to show the efficacy of F-measure in the case of a large-scale measurement, the error coefficients for different cluster levels L were examined as functions of time. In this part of the study, $1/L$ of the original measurement results were selected at regular

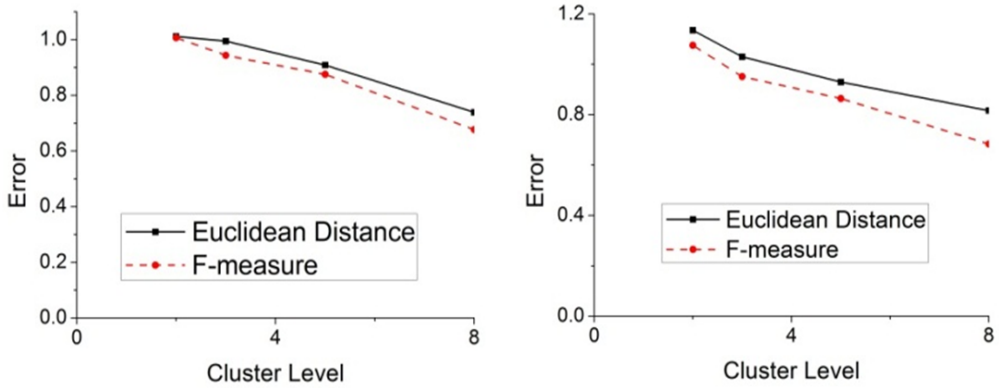

FIGURE 5. Comparison of error coefficients for Euclidean distance and F-measure. Left: Fpre = linear regression; Right: Fpre = decision tree.

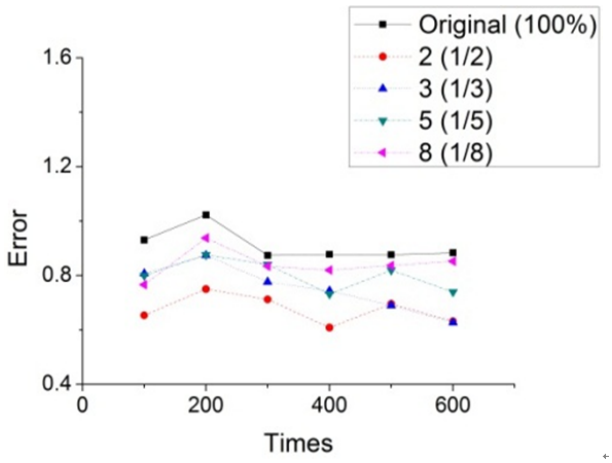

FIGURE 6. Error coefficients when cluster level is L with Fpre = decision tree

intervals. With this selected data, the experiment was repeated and the error coefficients were calculated. Figure 6 shows the changes in the error coefficients with increased "user access". Note that these results were compared with the error obtained for forecasting without clustering beforehand.

It is apparent from Figure 6 that, when the measurement scale is reduced to the $1/L$th original scale, the error coefficients may not increase in the case of clustering with the algorithms described above.

Figure 7 shows how much the measurement-scale is reduced by the clustering method. Error ratio means the ratio between the average error coefficients of the method without cluster and the error of cluster method. It represents how much error this experiment reduces compared with the original method. Error ratio = 100% means this experiment has the same error with the original method without any cluster. Figure 7 shows that, cluster with F-measure can reduce the measurement scale to the $1/L$th original scale without increasing the forecast error. No matter what the cluster level is, F-measure yields smaller error coefficients than Euclidean distance in this experiment. When measurement time has been reduced 7/8 of the original, the error has been decreased to 92% of the original error.

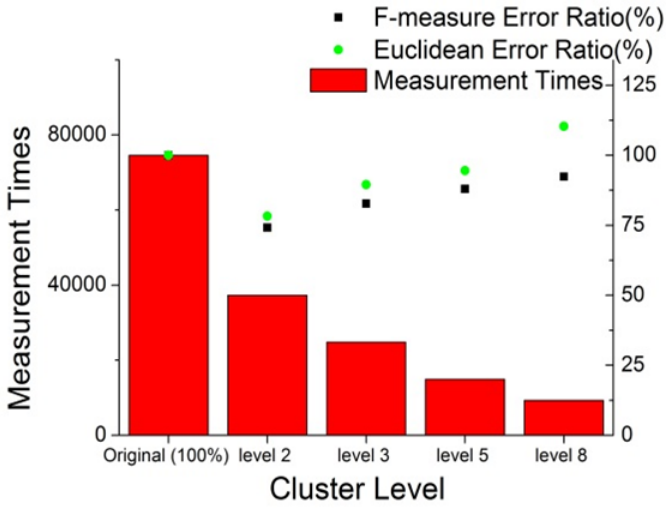

FIGURE 7. Measurement times when cluster level is L with Fpre = decision tree

In summary, F-measure yields smaller error coefficients with different basic forecasting algorithms and different cluster levels than the Euclidean distance. Thus, the proposed cluster-based measurement approach for large-scale video service performance, MCS, which employs F-measure, can reduce the number of measurements without increasing the forecast error. In experiment, this approach can reduce the measurement scale to the $1/L$th original scale. In addition, this approach can provide a large result ratio for forecasts when the measured values are sparse.

6. **Conclusions.** In this paper, a measurement approach based on service performance clustering for video services was proposed. The main contribution of this paper is the development of a forecasting-led time series similarity measure, which works well on sparse data with varying time intervals. This measure is sensitive to time displacement and does not require that the time series have the same timestamps. Importantly, this measure can reduce the number of measurements without increasing the prediction error of the overall measurement approach. Thus, the cost of large-scale measurement is reduced significantly, and the additional network pressure caused by measurement is also decreased. According

to the experiments conducted in this study, this approach reduces the workload of large-scale measurement to the $1/L$th original scale in active measurement, when the cluster level is L. Thus, it becomes possible for large-scale continuous online video measurement to generate low interference. The outcome of this work has, therefore, improved the feasibility of active measurement of large-scale online video services. In future work, some studies about the forecasting algorithm will combine with this measurement approach.

Acknowledgment. This work is supported by Special Fund for Strategic Pilot Technology of Chinese Academy of Sciences under Grant No. XDA06040602 and Youth Innovation Promotion Association of the Chinese Academy of Sciences No. Y529111601.

REFERENCES

[1] C. Labovitz, S. Iekel-Johnson, D. Mcpherson, J. Oberheide and F. Jahanian, Internet inter-domain traffic, *Proc. of the ACM SIGCOMM 2010*, vol.40, pp.75-86, 2010.

[2] China Internet Network Information Center (CNNIC), *The 41st China Statistical Report on Internet Development*, http://cnnic.cn/hlwfzyj/hlwxzbg/hlwtjbg/201801/P020180131509544165973.pdf, 2018.

[3] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan and H. Zhang, Understanding the impact of video quality on user engagement, *ACM SIGCOMM CCR*, vol.41, no.3, pp.362-373, 2013.

[4] S. S. Krishnan and R. K. Sitaraman, Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs, *IEEE/ACM Trans. Networking*, vol.21, no.6, pp.2001-2014, 2013.

[5] P. Gill, M. Arlitt, Z. Li and A. Mahanti, Youtube traffic characterization: A view from the edge, *Proc. of ACM SIGCOMM*, 2007.

[6] Z. Wang, L. Sun, C. Wu and S. Yang, Guiding Internet-scale video service deployment using microblog-based prediction, *Proc. of IEEE INFOCOM*, pp.2901-2905, 2012.

[7] V. K. Adhikari, Y. Guo, F. Hao and V. Hilt, Measurement study of Netflix, Hulu, and a tale of three CDNs, *IEEE/ACM Trans. Networking*, vol.23, no.6, pp.1984-1997, 2015.

[8] V. K. Adhikari, Y. Guo, F. Hao and M. Varvello, Unreeling netflix: Understanding and improving multi-CDN movie delivery, *Proc. of IEEE INFOCOM*, vol.131, pp.1620-1628, 2012.

[9] K. Lacurts, J. C. Mogul, H. Balakrishnan and Y. Turner, Cicada: Predictive guarantees for cloud network bandwidth, *Networking Machine Learning Traffic Prediction*, 2014.

[10] Z. Ye, A. Bouguettaya and X. Zhou, QoS-aware cloud service composition using time series, *Proc. of the 11th Int. Conf. Service-Oriented Comput.*, vol.8274, pp.9-22, 2013.

[11] Z. Ye, S. Mistry, A. Bouguettaya and H. Dong, Long-term QoS-aware cloud service composition using multivariate time series analysis, *IEEE Trans. Services Computing*, vol.9, no.3, pp.382-393, 2016.

[12] Y. Zhuo, J. You, J. Wang and H. Xue, Video measurement approach with classification based on service performance clustering, *ICCSN 2017*, pp.1240-1244, 2017.

[13] Y. Zhuo, J. You, J. Wang, W. Qi and N. Qiao, Measuring and commander system for online video service in sea service, *Computer Engineering*, http://kns.cnki.net/kcms/detail/31.1289.TP.20170512.1155.002.html, pp.1-8, 2017.

[14] C. C. Santos, J. Bernardes, P. M. B. Vitanyi and L. Antunes, Clustering fetal heart rate tracings by compression, *IEEE CBMS 2006*, pp.685-690, 2006.

[15] A. Bagnall and G. Janacek, Clustering time series with clipped data, *Machine Learning*, vol.58, no.2, pp.151-178, 2005.

[16] P. P. Rodrigues, J. Gama and J. P. Pedroso, Hierarchical clustering of time-series data streams, *IEEE Trans. Knowl. Data Eng.*, vol.20, no.5, pp.615-627, 2008.

[17] M. Valk and A. Pinheiro, Time-series clustering via quasi U-statistics, *J. Time Ser. Anal.*, vol.33, no.4, pp.608-619, 2012.

[18] L. N. Ferreira and L. Zhao, Time series clustering via community detection in networks, *Inf. Sci.*, vol.326, pp.227-242, 2016.

[19] P. Berkhin, *A Survey of Clustering Data Mining Techniques: Grouping Multidimensional Data*, Springer Heidelberg, Berlin, 2006.

[20] T. W. Liao, Clustering of time series data – A survey, *Pattern Recognition*, vol.38, no.11, pp.1857-1874, 2005.
[21] B. K. Yi and C. Faloutsos, Fast time sequence indexing for arbitrary L_p norms, *Proc. of the 26th International Conference on Very Large Databases*, Morgan Kaufmann Publishers Inc., pp.385-394, 2000.
[22] E. J. Keogh and M. J. Pazzani, *Derivative Dynamic Time Warping*, 2001.
[23] J. Shen, W. Huang, D. Zhu and J. Liang, A novel similarity measure model for multivariate time series based on LMNN and DTW, *Neural Processing Letters*, vol.45, no.3, pp.1-13, 2017.
[24] B. Boucheham, Reduced data similarity-based matching for time series patterns alignment, *Pattern Recognit. Lett.*, vol.31, no.7, pp.629-638, 2010.
[25] L. Chen and R. Ng, On the marriage of L_p-norms and edit distance, *Proc. of the 30th International Conference on Very Large Databases*, VLDB Endowment, pp.792-803, 2004.
[26] J. Lin and Y. Li, Finding structural similarity in time series data using bag-of-patterns representation, *Proc. of International Conference on Scientific and Statistical Database Management*, New Orleans, LA, USA, pp.461-477, 2009.
[27] G. E. Batista, E. J. Keogh, O. M. Tataw and V. M. Souza, CID: An efficient complexity-invariant distance for time series, *Data Min. Knowl. Discov.*, vol.28, no.3, pp.634-669, 2014.
[28] K. Y. Staroverova and V. M. Bure, Characteristics based dissimilarity measure for time series, *Vestnik St. Petersburg University, Mathematics*, vol.13, no.1, pp.51-60, 2017.
[29] Y. Xiong and D. Y. Yeung, Time series clustering with ARMA mixtures, *Pattern Recognition*, vol.37, no.8, pp.1675-1689, 2004.
[30] J. R. Quinlan, C4.5: Programs for machine learning, *DBLP: Computer Science Bibliography*, 1993.