# Enhancing Internet-scale Video Service Deployment Using Microblog-based Prediction

Zhi Wang, *Member, IEEE,* Lifeng Sun, *Member, IEEE,* Chuan Wu, *Member, IEEE,* and Shiqiang Yang, *Senior Member, IEEE*

**Abstract**—Online microblogging has been very popular in today's Internet, where users follow other people they are interested in and exchange information between themselves. Among these exchanges, video links are a representative type on a microblogging site. The impact is fundamental — not only are viewers in a video service directly coming from the microblog sharing and recommendation, but also are the users in the microblogging site representing a promising sample to all the viewers. It is intriguing to study a proactive service deployment for such videos, using the propagation patterns of microblogs. Based on extensive traces from Youku and Tencent Weibo, a popular video sharing site and a favored microblogging system, we explore how video propagation patterns in the microblogging system are correlated with video popularity on the video sharing site. Using influential factors summarized from the measurement studies, we further design a neural network-based learning framework to predict the number of potential viewers and their geographic distribution. We then design proactive video deployment algorithms based on the prediction framework, which not only determines the upload capacities of servers in different regions, but also strategically replicates videos to these regions to serve users. Our PlanetLab-based experiments verify the effectiveness of our design.

**Index Terms**—Online microblogging, video streaming, video service deployment

## 1 Introduction

Recent years have witnessed the blossom of microblogging services in the Internet, *e.g.*, Twitter, Google+, Plurk. In a microblogging system, users can create and maintain social connections among each other, as well as subscribe to contents shared by others from external content sharing system, as followers [1]. Among the variety of contents to exchange, links to videos on video sharing sites are a popular type — users from microblogging exchanges are constituting a large portion of viewers in YouTube-like video sharing sites [2]. Popularity patterns of such *socialized* videos have greatly changed as follows: (1) video popularity is highly effected by the online social networks; (2) the popularity becomes more instantaneous [3]. These changes make traditional popularity-based approaches for video service deployment suboptimal, if not completely ineffective [4].

Since a microblogging system is closely connected to many content sharing sites, it ideally samples valuable information about how users produce and share contents from those sites. Video propagation models in a microblogging system can be exploited for a better prediction of video popularity patterns, so as to improve the service quality of a video sharing system In this paper, we advocate to exploit the sampling and prediction capabilities of a microblogging system to provide better Internet video services.

In a typical video sharing site today, large volumes of videos are uploaded by users, with viewers from all over the world. In 2013, more than 100 hours' worth of videos were uploaded every minute in YouTube, serving up to 1 billion unique viewers every month [5]. A common practice to provide these video services is to replicate videos in servers at different geographic locations [6], but it is impractical to replicate all the videos in every location. An effective and adaptive replication strategy to serve the dynamic demand for different videos in different geographic regions, is in need.

In this paper, we propose to exploit video sharing patterns from a microblogging system for this purpose. The potential benefits are two-fold: (1) a video sharing site typically has no information about how video views propagate among its users, while a view propagation model could enable more effective view prediction; (2) the exchanges of video links in a microblogging system typically happen earlier than the actually video views on a video sharing site, and the time lag between both events can allow more timely and proactive deployment of videos. Based on our preliminary findings of the connection between the popularity of a video and how the video is shared in a microblogging system [7], in this paper, we focus on employing video microblog propagation patterns to improve the deployment of video services. Our contributions can be summarized as

*Z. Wang is with the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China (email: wangzhi@sz.tsinghua.edu.cn). L. Sun and S. Yang are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China (e-mail: {sunlf,yangshq}@tsinghua.edu.cn).*

*C. Wu is with the Department of Computer Science, the University of Hong Kong, Hong Kong (e-mail: cwu@cs.hku.hk).*

*This work is supported in part by the National Basic Research Program of China (973) under Grant No. 2011CB302206, the National Natural Science Foundation of China under Grant No. 61133008, 61210008, and 61272231, the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, Beijing Key Laboratory of Networked Multimedia, and a grant from Hong Kong RGC under the contract HKU 717812E.*

follows.

▷ In Sec. 3, we explore connections between microblogging exchanges of video links and popularity of videos, based on extensive traces collected from Tencent Weibo (hereafter, Weibo, a Twitter-like Chinese microblogging system) and Youku (an Internet video sharing site with immense popularity in China). We identify important characteristics of Weibo, which influence video access patterns on Youku: (1) the number of users that have *imported* a video to Weibo, (2) the number of users which *re-share* links to the video to their followers, (3) the number of followers that the video link share can reach, and (4) the *geographic distribution* of Weibo users.

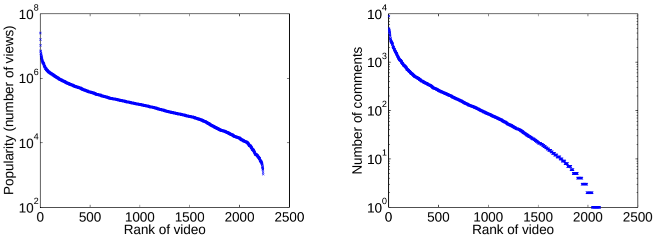
▷ In Sec. 4, we exploit these influential factors in the design of a neural network-based learning framework, for predicting the number of potential viewers of different videos and the geographic distribution of viewers. The accuracy of our prediction models is verified by trace-driven cross-validation experiments, as compared to a classical approach of linear regression-based forecasting.

▷ In Sec. 5, we further design a proactive video deployment algorithm based on the prediction frameworks. The algorithm can significantly improve the video download performance of users, according to our trace-driven experiments.

## 2 Related Work

Many architectures have been proposed to implement a large-scale video service, which distributes videos to users across the Internet as follows. (1) *Server-based strategies*, *e.g.*, the content distribution networks (CDNs) [8], which have powered today's dominating HTTP streaming. (2) *Client-based strategies*, *e.g.*, peer-to-peer content distribution were widely used in live video distribution and on-demand video distribution [9]. (3) *Hybrid strategies*, *e.g.*, a hybrid CDN and P2P distribution framework [10]. Traditional video distributions generally work in a passive way, in that video replication and cache are scheduled according to the video access patterns perceived by the servers or peers, *e.g.*, using linear regression approach to infer the future popularity of a video [11].

Video services in the Web 2.0 era focus more on the "social effects", including user experience, user participation and interaction with rich media. The impact of such social effects is fundamental, because of not only the huge amount of videos generated by users, but also the change of the video popularity distribution[12]. Li *et al.* [13] have studied the video sharing in the online social network, and observed the skewed popularity distribution of contents and the power-law activity of users. Traditional video distribution strategies designed without the consideration of such social influence, achieve sub-optimal performance in distributing videos in the context of online social network. Saxena *et al.* [14] have revealed that at some locations, the average service delay of YouTube used to be as large as $6.5$ seconds, due to the inefficient replication and distribution strategies [6], *e.g.*, the slow reaction to the popularity change.



(a) Number of views.

(b) Number of comments.

Fig. 1. Popularity of the sampled videos.

Only by carefully considering the social network information can the video sharing systems effectively distribute social/socialized video contents. Krishnamurthy *et al.* [15] have investigated an online microblogging system, Twitter, and identified distinct classes of Twitter users and their behaviors, as well as geographic growth patterns of the social network. With increasing popularity, a microblogging system resembles the real society. Interests, beliefs, and behavior of users in a microblogging system are representative of those in the real world [16]. To exploit the similarities, Ritterman *et al.* [17] advocated to forecast a swine flu pandemic based on a belief change model summarized from Twitter. In context of content popularity prediction based on a microblogging system, different models have been used for prediction in a variety of scenarios, including various linear regression models [18] and machine learning models [19]. Yang *et al.* [20] investigate the prediction of information diffusion in Twitter, in terms of the speed, scale, and range. Szabo *et al.* [21] study how to predict the popularity of contents on Digg and YouTube, using their own historical popularities. To the best of our knowledge, we are not aware of any existing study exploiting characteristics of a microblogging system to predict video access in another *external* video sharing system.

## 3 Measurements and Analysis

### 3.1 Collection of Traces

We have obtained traces from Youku and Weibo as follows.

**Youku**. In our study, we collected $2,291$ representative videos from $5$ popular categories on Youku, including "Music", "News", "Entertainment", "Baby" and "Original". As video sharing systems usually do not share detailed video popularity information, we crawled the view numbers of the videos periodically, so as to study their popularity change over time. The crawling was carried out during June 20 to June 30, 2011, on an hourly basis to avoid being blocked from frequent crawling. Each trace log indicates the cumulative number of views of each video since when the video was published until

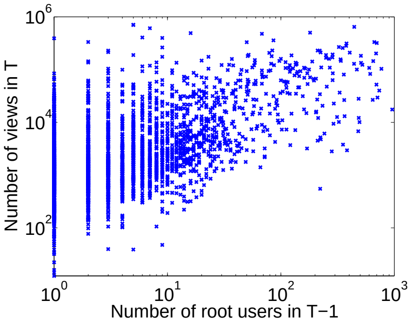


Fig. 2. The number of views vs. the number of root users in the previous time slot.

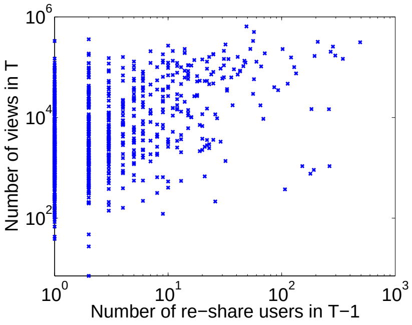


Fig. 3. The number of views vs. the number of re-share users in the previous time slot.

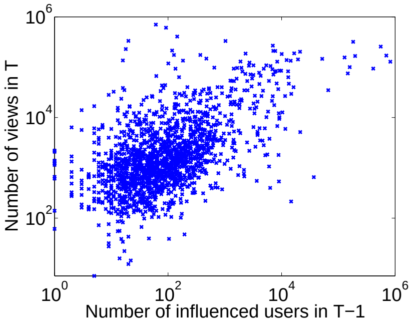


Fig. 4. The number of views vs. the number of influenced users in the previous time slot.

when the log was recorded, based on which we can calculate the average number of views of a video in each hour.

These $2,291$ videos cover a large variety of popularities, in terms of the number of views and comments. As illustrated in Fig. 1(a), each sample represents the total number of views of a video since its publication until the end of the ten days, versus the rank of the video. In Fig. 1(b), each sample represents the total number of comments posted by users (Youku allows users to post comments to videos) starting from its publication to the end of the ten days, versus the rank of the video. We observe that our dataset contains representative videos in a sense that they cover a large variety of popularities.

**Tencent Weibo**. We obtained Weibo traces from the technical team of Tencent, containing valuable runtime data of the system in the entire span of June 2011. Each entry in the traces corresponds to one microblog published, including ID, name, IP address of the publisher, time stamp when the microblog was posted, IDs of the parent and root microbloggers if it is a re-post, and contents of the microblog. We parsed the traces and obtained $4,468,398$ microblogs related to the videos above, *i.e.*, the microblogs that either contain the links to the videos or are re-shares of the microblogs that contain the links. In our study, we also used the social relationship between users involved in these microblogs.

### 3.2 View Number Predictability

To investigate the correlation between video link propagation on Weibo and the actual number of viewers in Youku, we study the following measurements in Weibo traces.

**Number of Views and Number of Root Users**. On Weibo, different users may introduce links to the same video on Youku from time to time, each of whom becomes a *root user* in this video's propagation. The more root users a video has, the more likely the video can attract more views in the future. With all the samples we collected, a Pearson's sample correlation coefficient [22] of $0.31$ is computed from the pairs of the numbers of root users and the numbers of views at a lag of $1$ time slot (in this paper, each time slot is $4$ hours) for these videos, showing positive correlation between the two quantities, as illustrated in Fig. 2.

**Number of Views and Number of Re-share Users**. Similarly, when more users are re-sharing (referred to as *re-share users*) the links to their followers, the more views can be expected on Youku. We have computed a Pearson's sample correlation coefficient of $0.29$ between the pairs of the numbers of re-share users and the numbers of views at a lag of $1$ time slot. Again, positive correlation is observed between the two quantities, as illustrated in Fig. 3.

**Number of Views and Number of Influenced Users**. The *influenced users* on Weibo (followers of root and re-share users of a video link, who can see the microblogs of the video) may likely become actual viewers on Youku themselves. Specifically, a Pearson's sample correlation coefficient of $0.15$ is derived from the pairs the numbers of influenced users and the numbers of views at a lag of $1$ time slot, as illustrated in Fig. 4.

### 3.3 Geographic Distribution Predictability

For video service deployment, we also need information about geographic distribution of viewers of different videos. Since Weibo users sharing a video link are "samples" of all viewers of that video on Youku, we investigate the geographic distribution of Weibo users who have published a microblog containing a link to the video, and use it to estimate the distribution of all viewers in Youku. The rationale is that such microblogs are published by root and re-share users of the video, who may well have just viewed the video before posting the microblogs. Our design in this paper, however, is not limited to the Weibo sample of viewers' geographic distribution.

Given that the majority of viewers of Youku videos are in China, we consider $5$ representative regions in China, namely BJ (Beijing), SH (Shanghai), SZ (Shenzhen), CD (Chengdu) and XA (Xi'an), where large CDNs in China commonly deploy data centers [23], and an overseas region, referred to as OS (overseas). We use $\mathcal{R}$ to denote the set of these regions, *i.e.*, $\mathcal{R} = \{BJ, SH, SZ, CD, XA, OS\}$.

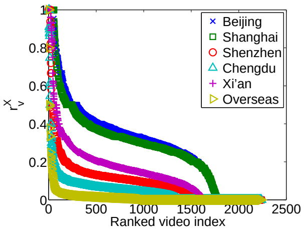


Fig. 5. Geographic distribution of Weibo users involved in propagation of different videos.

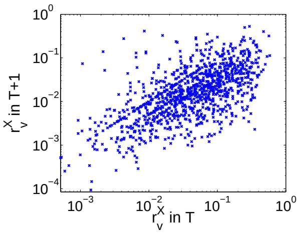


Fig. 6. Correlation between fractions of Weibo users at a region in two consecutive time slots.

We map the IP addresses of users in our Weibo traces to the six regions using an IP-to-location mapping database, and estimate the geographic distribution of viewers of video $v$ at time $T$ by a 6-dimensional vector

$$G_v^T = \{r_v^{BJ}(T), r_v^{SH}(T), r_v^{SZ}(T), r_v^{CD}(T), r_v^{XA}(T), r_v^{OS}(T)\},$$

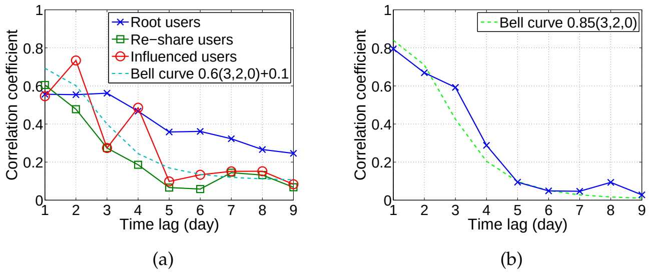
where $r_v^X(T)$ is the normalized fraction of Weibo microblogs containing links to video $v$, posted by users in region $X$ in time slot $T$, and $r_v^{BJ}(T)+r_v^{SH}(T)+r_v^{SZ}(T)+r_v^{CD}(T)+r_v^{XA}(T)+r_v^{OS}(T)=1$.

**Skewed Geographic Distribution.** Fig. 5 plots the average fraction of microblogs posted in each region containing links to each of the videos, among all the microblogs posted for all videos in the ten-day trace span. We observe that the distribution over different regions is highly skewed: as large as $40\%$ of viewers of over $50\%$ of the videos reside in Beijing and Shanghai regions, while very small fractions of viewers are from the overseas. This observation indicates that heterogeneous video service deployment is needed for different regions.

**Predictability of Future Geographic Distribution.** To investigate whether future geographic distribution can be predicted by historical distributions, we plot in Fig. 6 the fraction of microblogs posted in a region in time slot $T$ (*i.e.*, $r_v^X(T), X \in \mathcal{R}$), versus that in the previous time slot $T-1$ (*i.e.*, $r_v^X(T-1)$) , for each of the regions ($X \in \mathcal{R}$) and each of the videos ($v \in \mathcal{V}$). Positive correlation between the two can be observed, with a correlation coefficient of 0.29. This observation suggests that historical geographic distribution can potentially predict the future distribution.

### 3.4 Impact of Measures at Different Time Lags

Besides observing correlations between numbers of root/re-share/influenced users (*resp.* fraction of Weibo users of a video) in time $T-1$ and Youku view numbers (*resp.* fraction of Weibo users of a video) at $T$, we further investigate the correlation at different time lags between the two. To avoid the impact of videos that are only viewed in a very short time span, we have selected 100 videos that have a relatively long popularity span, *i.e.*, they were regularly viewed by users in 10 days.



(a)

(b)

Fig. 7. Influence of the numbers of root/re-share/influenced users on the view number and geographic distribution at different time lags.

First, for each of the 100 selected videos, we calculate the Pearson correlation coefficient between a 10-day series of the number of views of the video on Youku and a 10-day series of the number of root/re-share/influenced users of the video, at different time lags between the two series, respectively. In Fig. 7(a), each sample represents the average correlation coefficient over those of all videos, calculated at a specific time lag between the involved time series. We see that the correlation weakens as the time lag becomes larger, and the correlation coefficients are quite small when the lag is larger than 7 days. Hence, we will use measurements collected in the previous 7 days for the prediction of view numbers only.

Second, we plot in Fig. 7(b) the Pearson correlation coefficients between the fractions of a video's microbloggers in the six regions at $T$ and those at different time lags, where each sample is the average over those of all 100 videos with data extracted from a 10-day interval. Similarly, recent geographic distributions have more influence on the future geographic distribution, which will be weighted more in our prediction model in Sec. 4.

## 4 Neural Network for View Prediction

When making video service deployment decisions, the video service provider is interested to know two things about a video in the near future: (1) Will the video attract more or fewer viewers, so that more or fewer servers should be allocated to serve the video? (2) Servers in which regions should be used, so as to best satisfy these users from different locations?

### 4.1 Using Neural Network as Prediction Model

In our design, we predict the number of views of a video, and the geographic distribution of the viewers, based on historical information from the microblogging system, using neural networks.

**Merits of using neural networks.** (1) Neural networks have been proven effective for time series prediction [24], as what we are pursuing. (2) According to our measurement study, the relationship between the number of root/re- share/influenced users and the number of

Youku views can be non-linear. Neural networks are effective for learning such non-linear relationship [25].

**The structure of the neural networks.** In our study, we train two neural network models, (A) one for the prediction of the total number of views of a video, and (B) the other for forecasting the geographic distribution of viewers. We use a 3-layer feed-forward neural network for both predictions, since it has been proven that multilayer neural networks with only one hidden layer are universal approximators [26]. In predicting video views, the structure of the neural network may remain for a relatively long time, but the training of the networks may take place frequently, using the new training dataset in the most recent time slots, as long as the video service provider has enough computation resource.

## 4.2 Constructing Input Features

To use the social measurements in Sec. 3 as input features, we need to decide a time window, a frequency for feature exaction in the time window (*i.e.*, the number of time slots sampled in the time window), and the weight of each feature in the learning framework.

**Time Window.** We use a proper time window to avoid noisy features to be selected for the training. According to Sec. 3.4, the correlation between the number of Youku views and its influential Weibo measurements (*i.e.*, the numbers of root/re-share/influenced users), as well as the correlation between the geographic distribution of viewers and its influential measurements (*i.e.*, the historical fraction of users residing in different regions), are weaker when the time lags are larger. Hence, we only extract features within the recent 7 days to train neural networks (A) and (B). For a newly published video with a lifetime shorter than 7 days, we use measurements throughout its past lifetime.

**Frequency.** The features are extracted from the following time slots: $T-1, T-7, T-13, \ldots$, *i.e.*, consecutive features are collected with a time interval of $24$ hours (recall that each time slot is $4$ hours), to capture the daily patterns. Let $M$ denote the number of days the features are extracted.

**Weight.** Existing studies have shown that the learning performance of a neural network model can be improved by properly weighting the input features [27]. We weight the features from different time slots according to their levels of correlation with the prediction targets. In Fig. 7(a) and (b), the curves of Pearson correlation coefficients can be fitted well by generalized bell functions $f(x) = \frac{e}{1+\left|\frac{x-c}{a}\right|^{2b}} + d$ (in Fig. 7, for simplicity, we denote a particular bell function by "$e(a,b,c)+d$"). Hence, we weight the number of root/re-share/influenced users, to be used as features in neural network (A), by $\alpha(x) = \frac{0.6}{1+|x/3|^4} + 0.1$, and the past geographic distributions of viewers, which are used as features in neural network (B), by $\beta(x) = \frac{0.85}{1+|x/3|^4}$, where $x$ is the time lag between the time slots when the prediction target and the corresponding features happen, respectively.

In our design, we use $R_v^T(i)$, $S_v^T(i)$, and $I_v^T(i)$ to denote the number of root, re-share, and influenced users of video $v$ in the $i$th time slot in the time window before $T$, respectively. We use $G_{v,r}^T(i)$ to denote the fraction of microblogs of video $v$ posted by users in region $r$ in the $i$th time slot in the time window before $T$.

Let $X_v^T$ and $Y_v^T$ be the features of the samples for training neural network (A) and neural network (B), respectively. We have $X_v^T = \{\alpha(i)R_v^T(i), \alpha(i)S_v^T(i), \alpha(i)I_v^T(i) | i = 1, 2, \ldots, M\}$ in neural network (A), and features $Y_v^T = \{\beta(i)G_{v,r}^T(i) | \forall r \in \mathcal{R}, i = 1, 2, \ldots, M\}$ in neural network (B).

## 4.3 Samples for Training and Evaluation

To train and evaluate the neural networks, we first pre-labeled samples from the traces, and each sample consists of the input features and the prediction target(s). In neural network (A), a sample is $\{X_v^T, \bar{V}_v^T\}$, where $\bar{V}_v^T$ is the vector denoting the level of view number of video $v$ in time slot $T$. Using the level of view number is sufficient for bandwidth allocation, and can provide much better prediction accuracy than using the exact view number. According to the popularity distribution of the videos illustrated in Fig. 1(a), we classify the number of views $N$ for a video into 5 levels: (1) $N < 500$, (2) $500 \leq N < 5000$, (3) $5000 \leq N < 10000$, (4) $10000 \leq N < 100000$, and (5) $N \geq 100000$. $\bar{V}_v^T$ is a 5-dimensional binary vector with $\bar{V}_v^T[i] = 1$ and $\bar{V}_v^T[j] = 0, \forall j \neq i$, denoting that the number of views belongs to the $i$th level. The rationale of classifying the view numbers into different levels is that in video service deployment, the level of view numbers determines the bandwidth resource needed.

On the other hand, in neural network (B), a sample is $\{Y_v^T, G_v^T\}$, where $G_v^T$ is a 6-dimensional vector in which each element represents the fraction of microblogs of the video posted in one of the six regions in $\mathcal{R}$ in time slot $T$, respectively.

In summary, we totally have $35,000$ samples for the neural network training and evaluation, corresponding to $1000$ videos, covering both old videos and newly published ones. We randomly use $6,000$ of them for the evaluation, and the other $29,000$ for the training process.

## 4.4 Training Neural Networks

**Training Neural Network (A).** The *output layer* in the neural network for predicting the number of views, consists of 5 neurons, corresponding to the elements in vector $\bar{V}_v^T$, respectively. The *input layer* has $3M$ nodes, corresponding to the feature set $X_v^T$, *i.e.*, $M = 7$ for old videos, and $M = 1, 2, \ldots, 6$ for new videos published $M$ days ago. In the *hidden layer*, the number of neurons is decided as follows: we vary the number of hidden neurons from $15$ to $25$, and measure the number of samples whose view numbers can be classified into the correct levels, using a validation set containing 25% of all samples from the training set (*i.e.*, $7,250$ samples out of $29,000$). We observe that $15$ hidden neurons can achieve the best results in cases of old videos with a lifetime longer than 7 days, and $18$ hidden neurons are needed for most of the new videos.

**Training Neural Network (B).** The output layer in the neural network for predicting the geographic distribution of viewers corresponds to the 6-dimensional vector $G_v^T$. The input layer corresponds to a $6M$-dimensional vector, containing $M$ vectors of viewer geographic distributions (each element is the fraction of microbloggers of video $v$ in each of the six regions) in the previous $M$ days. To determine the number of neurons in the hidden layer, we measure the MSE (mean squared error) between the output geographic distribution vector and the actual geographic distribution vector. A smaller MSE indicates that the predicted geographic distribution vectors are more accurate, *i.e.*, the differences between the actual geographic distribution vectors and them are smaller. Neural network (B) for geographic distribution prediction is trained using the same traces as used in training Neural network (A), and our training results give that $20$ hidden neurons provide the best accuracy for old videos, and $35$ hidden neurons for new videos. Next, we will evaluate the accuracy of both the neural network models.

### 4.5 Evaluating the Predication Accuracy

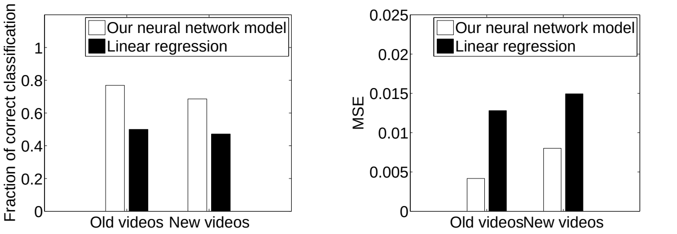
We evaluate the accuracy of our neural network models using the $6000$ samples. We compare the accuracy of our neural networks with that of a linear regression approach [11], in which the parameters (*i.e.*, $\gamma, \epsilon$) in the linear model (*i.e.*, $y = \gamma x + \epsilon$), are calculated based on minimizing the squared error of the $M$ samples — the view numbers or user geographic fractions (*i.e.*, $y$), and time points (*i.e.*, $x$) as used for training samples in our neural network models.

Fig. 8 shows the evaluation results. We observe that our neural network models achieve much better prediction accuracy than that achieved by the linear regression approach. In addition, the number of views and geographic distribution of old videos can be better forecasted, than those of the new videos, due to the fewer number of features in the learning framework for new videos. These results indicate that in today's social video sharing, using social information which reflects how videos are shared among users through social connections, for predicting video popularity is promising.

## 5 Enhanced Video Service Deployment

### 5.1 Deployment Scheme and Objective

A distributed video service platform involves data centers in different geographic regions. In each data center, there is a shared storage backbone that stores the deployed videos, and streaming servers that upload the videos to the requesting viewers. Better Internet connectivity and upload bandwidth can be achieved when a user downloads the video from a server located at a region that is close to the user [28].



(a) Number of views

(b) Geographic distribution of viewers

Fig. 8. Prediction accuracy: a comparison.

The objective of our deployment algorithms is to proactively decide the total amount of upload bandwidth to reserve in each data center for this video service, which videos to be replicated in each data center at each time, and how to schedule the upload to viewers of different videos from different regions, so that little video download delay is experienced by the viewers in the entire system.

Let binary vector $F_v^T = \{f_{v,r}^T | \forall r \in \mathcal{R}\}$ denote the replication plan for video $v$ in time slot $T$, where $f_{v,r}^T = 1$ indicates that video $v$ is stored in region $r$ in $T$ and $f_{v,r}^T = 0$ otherwise. We use $c_{i,r}$ to denote the gain for a user in region $i$ to download a video from servers in region $r$, which is calculated as the inverse proportion of the average delay between regions $i$ and $r$, *i.e.*, $c_{i,r} = \frac{1}{RTT(i,r)}$, where $RTT(i,r)$ is the average round-trip time (RTT) between the two regions (other gain function can also apply in our design, *e.g.*, end-to-end bandwidth, *etc.*). When downloading a video, a viewer in region $i$ first tries to request it from servers with the smallest RTT, so as to achieve the largest gain. If servers in that region are not able to serve this request, the viewer's request is redirected to the next best region, until the request is eventually served. On the other hand, a streaming server serves all the requests when it has enough upload bandwidth; otherwise, it selects the requests to serve, by prioritizing those from regions with small RTTs to the region where the server is located.

### 5.2 Regional Upload Bandwidth Reservation

Video upload bandwidth in each region is purchased from the respective Internet service providers. Let $W$ denote the total budget that the video service provider is willing to spend on the upload capacity. We decide the amount of upload bandwidth to reserve in each region according to the average number of concurrent requests $V_r, \forall r \in \mathcal{R}$, and the unit prices $P_r, \forall r \in \mathcal{R}$, for upload bandwidth in different regions. Suppose that every video request is served by one unit upload bandwidth.

The upload bandwidth reservation in each region is carried out every $L$ time slots, considering the practice that bandwidth reservations are made for long periods in the real world. We compute $V_r$ as the average number of concurrent requests from region $r$ in the previous $L$ time slots. We determine the bandwidth reservation by solving the following optimization problem:

$$\max_{\mathcal{U}} \sum_{r \in \mathcal{R}} z_r(U_r) \tag{1}$$

subject to:

$$\sum_{r \in \mathcal{R}} P_r U_r \le W, \quad \text{and} \quad U_r \ge 0, \forall r \in \mathcal{R},$$

where $\mathcal{U} = \{U_r, \forall r \in \mathcal{R}\}$, and $U_r$ is the upload capacity reserved in region $r$. $z_r(U_r)$ is the gain for reserving upload capacity $U_r$ in region $r$. Suppose $c_{r_0,r} \ge c_{r_1,r} \ge c_{r_2,r} \ldots \ge c_{r_{R-1},r}$, where $r_0 = r$ and $r_i \in \mathcal{R} \setminus \{r\}, i = 1, \ldots, R-1$ (Recall $c_{i,r} = \frac{1}{RTT(i,r)}$). $z_r(U_r)$ is defined as follows:

$$z_r(U_r) = \begin{cases} c_{r_0,r} U_r, \text{ if } 0 \le U_r \le V_{r_0}; \\ z_r(V_{r_0}) + c_{r_1,r}(U_r - V_{r_0}), \text{ if } V_{r_0} < U_r \le V_{r_0} + V_{r_1}; \\ \ldots \\ z_r(\sum_{i=0}^{R-2} V_{r_i}) + c_{r_{R-1},r}(U_r - \sum_{i=0}^{R-2} V_{r_i}), \\ \quad \text{if } \sum_{i=0}^{R-2} V_{r_i} < U_r \le \sum_{i=0}^{R-1} V_{r_i}; \\ z_r(\sum_{i=0}^{R-1} V_{r_i}), \text{ if } \sum_{i=0}^{R-1} V_{r_i} < U_r. \end{cases}$$

The rationale is as follows: When the reserved upload capacity in region $r$ is no larger than the amount needed for serving all requests in the region, all the reserved bandwidth in $r$ is to be used to serve only the requests from $r$, achieving a gain of $c_{r,r}U_r$. When the reserved upload capacity in region $r$ is more than that needed for serving this region, the extra bandwidth is first to serve requests from region $r_1$ with the largest $c_{r_1,r}$ (*i.e.*, smallest delay to $r$), achieving an additional gain of $c_{r_1,r}(U_r - V_r)$; if there is further bandwidth left, it can be used to serve requests from region $r_2$, and so on.

We design Algorithm 1 to solve the optimal bandwidth reservation problem in (1): We always allocate a certain amount of upload bandwidth ($\Delta$) to region $r'$ with the current largest positive marginal gain per unit price $z'_r(U_r)/P_r$, where the marginal gain $z'_r(U_r)$ is derived as follows:

$$z'_r(U_r) = \begin{cases} c_{r_0,r}, & \text{if } 0 \le U_r \le V_{r_0}, \\ c_{r_1,r}, & \text{if } V_{r_0} < U_r \le V_{r_0} + V_{r_1}, \\ \ldots \\ c_{r_{R-1},r}, & \text{if } \sum_{i=0}^{R-2} V_{r_i} < U_r \le \sum_{i=0}^{R-1} V_{r_i}, \\ 0, & \text{if } \sum_{i=0}^{R-1} V_{r_i} < U_r. \end{cases}$$

The amount of upload bandwidth to allocate to region $r'$ is computed by $\Delta = \min(\sum_{i=0}^{k} V_{r'_i} - U_{r'}, W'/P_{r'})$, where $\sum_{i=0}^{k-1} V_{r'_i} \le U_{r'} < \sum_{i=0}^{k} V_{r'_i}$, and $W'$ denotes the remaining budget. ${r'}_i, i = 0, 1, \ldots$, are the series of regions such that $c_{r'_0,r'} \ge c_{r'_1,r'} \ge c_{r'_2,r'} \ge \ldots$. Region $k$ is the one selected from the ranked regions whose viewing requests are to be served by servers in region $r'$, to achieve the largest gain. Then we deduct $P_{r'}\Delta$ from the budget $W'$ and repeat the allocation until all the budget is used up or the upload capacities allocated can serve all the requests already.

**Algorithm 1** Upload bandwidth reservation for different regions (executed every $L$ time slots)

```
procedure UploadAllocation($W, P_r, c_{r_i,r}, i = 1, \ldots, R-1, \forall r \in \mathcal{R}$)
    Update $V_r \forall r \in \mathcal{R}$ as the average number of concurrent video requests
    $U_r \leftarrow 0, \forall r \in \mathcal{R}$
    $W' \leftarrow W$
    while $W' > 0$ and $\exists r \in \mathcal{R}, {z'}_r(U_r) > 0$ do
        Choose the region $r'$ with the largest $z'_r(U_r)/P_r$ among all the regions
        if $\exists k, \sum_{i=0}^{k-1} V_{{r'}_i} \le U_{r'} < \sum_{i=0}^{k} V_{{r'}_i}$ then
            $\Delta \leftarrow \min(\sum_{i=0}^{k} V_{{r'}_i} - U_{r'}, W'/P_{r'})$
            $U_{r'} \leftarrow U_{r'} + \Delta$
            $W' \leftarrow W' - P_{r'}\Delta$
        end if
    end while
end procedure
```

The complexity of Algorithm 1 depends on the number of regions to allocate the upload bandwidth resource to. In particular, the algorithm iteratively allocates the upload bandwidth to the "slots" in each region, where a slot represents the amount of bandwidth allocated in each iteration. Thus, the time complexity of this algorithm is $O(|\mathcal{R}|^2 \log |\mathcal{R}|)$, where $\log |\mathcal{R}|$ is a result of maintaining a heap structure for the ranking. Since there are only tens of regions deployed by a large video service provider, this algorithm can be carried out effectively even in a centralized manner.

### 5.3 Video Replication

Next, we use the following heuristic algorithm to replicate videos to these regions, as summarized in Algorithm 2.

**(i) Predicting views using neural network models.** The video service provider collects statistics from the online microblogging system, *i.e.*, the number of root/reshare/influenced users and the geographic distribution of video microblogs in the previous $M$ days, and estimates the level of views, $\bar{V}_v^T$, and the geographic distribution of views, $G_{v,r}^T$, of each video from each region in the next time slot, using our proposed neural networks. Note that the neural network models are calibrated at the end of each time slot, based on the newly collected view numbers and distribution.

Let $V_{v,r}^T$ denote the predicted number of concurrent video requests from region $r$ for video $v$ in time slot $T$, which is derived by $V_{v,r}^T = avg(\bar{V}_v^T) G_{v,r}^T \frac{t_v}{t_t}$, where $avg(\bar{V}_v^T)$ is the average number of views in output level $\bar{V}_v^T$. We assume that the video popularity is uniformly distributed in the popularity groups (1–4), and the average number is then calculated as follows, $avg(1) = 250$, $avg(2) = 2750$, $avg(3) = 7500$, $avg(4) = 55000$, and for the last popularity group 5, we let $avg(5) = 150000$, where $avg(5)$ corresponds to videos with the number of views larger than $100000$ – there are very few videos with view number larger than $100000$ and the average number is closer to $150000$, as illustrated in Fig. 1(a). $t_v$ is the average service time for a request of video $v$ and $t_t$ is the time slot length ($t_v << t_t$, since $t_v$ is generally at minutes for short videos and $t_t$ is 4 hours). In doing so, we seek to use the number of concurrent video requests for bandwidth allocation (recall that each video request is served by one unit upload bandwidth).

**(ii) Replicating videos to different regions.** Let $q_v(i, j, F_v^T)$ denote the estimated number of concurrent viewing requests for video $v$ from region $i$ that will be sent to region $j$ under replication plan $F_v^T$ in $T$. According to the request service scheme discussed at the beginning of this section, $q_v(i, j, F_v^T)$ can be derived as follows:

$$q_v(i,j,F_v^T) = \begin{cases} 0, & \text{if } f_{v,j}^T = 0 \\ \begin{cases} 0 & f_{v,i}^T = 1, i \neq j \\ V_{v,i}^T & i = j \\ \frac{V_{v,i}^T}{\sum_{k\in\mathcal{R}} f_{v,k}^T} & f_{v,i}^T = 0 \end{cases}, & \text{if } f_{v,j}^T = 1 \end{cases}.$$

The rationale lies in the following: (1) if $v$ is not deployed in $j$ or $v$ is deployed in $i$ itself, where $i \neq j$, no request for video $v$ from region $i$ is to be sent to region $j$, *i.e.*, $q_v(i, j, F_v^T) = 0$; (2) if $i = j$ and the video is deployed in region $i$ ($j$), all requests are potentially served locally, *i.e.*, $q_v(i, j, F_v^T) = V_{v,i}^T$; and (3) if $v$ is not deployed in $i$ but deployed in $j$, each region caching the video receives an equal share of the requests from $i$, assuming users' region preference is uniformly distributed, *i.e.*, $q_v(i, j, F_v^T) = \frac{V_{v,i}^T}{\sum_{k\in\mathcal{R}} f_{v,k}^T}$. We calculate the gain of video $v$ under a particular deployment plan $F_v^T$ as follows:

$$B_v(F_v^T) = \sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{R}} c_{i,r} q_v(i, r, F_v^T).$$

To find out whether a video should be replicated to a particular region, we evaluate a marginal gain $b_{v,r}$, which represents the potential improvement in video $v$'s gain if $v$ is to be deployed in region $r$, than not deployed:

$$b_{v,r} = B_v({F'}_v^T) - B_v(F_v^T), \qquad (2)$$

where ${F'}_v^T = \{{f'}_{v,i}^T | \forall i \in \mathcal{R}\}$, and ${f'}_{v,i}^T = \begin{cases} f_{v,i}^T, & i \neq r \\ 1, & i = r. \end{cases}$

We first initialize $F_v^T \leftarrow \{0, 0, \ldots\}$, $\forall v \in \mathcal{V}$, so that videos will be deployed according to their requests in the current time slot. Let $\mathcal{B} = \{b_{v,r} | \forall v \in \mathcal{V}, \forall r \in \mathcal{R}\}$, and sort $\mathcal{B}$ in descending order. To best utilize the reserved bandwidth $U_r$ in each region $r$, each time we select a video-region pair with the largest $b_{v,r}$, deploy video $v$ in region $r$ in $T$ (or retain video $v$ in region $r$ in $T$ if it is already deployed) if $r$ still has enough upload capacity to serve all the requests for the video that will be sent to $r$, and remove $b_{v,r}$ from $\mathcal{B}$. If $v$ is deployed in $r$, $b_{v,l}, \forall l \in \mathcal{R}$ should be updated and $\mathcal{B}$ is resorted. The steps repeat until none of the regions has upload capacity left or there is no video-region candidate in set $\mathcal{B}$.

**Algorithm 2** Video Deployment based on Prediction.

```
1:  procedure VideoDeployment($\bar{V}_v^T, G_v^T, \forall v \in \mathcal{V}$)
2:      Initialize $F_v^T \leftarrow \{0, 0, \ldots\}$, $\forall v \in \mathcal{V}$
3:      Calculate $b_{v,r}, \forall v \in \mathcal{V}, \forall r \in \mathcal{R}$, according to Eq. (2)
4:      $\mathcal{B} \leftarrow \{b_{v,r} | \forall v \in \mathcal{V}, \forall r \in \mathcal{R}\}$
5:      Sort $\mathcal{B}$ in descending order
6:      $Q(v, r, F_v^T) \leftarrow \sum_{i\in\mathcal{R}} q_v(i, r, F_v^T), \forall v \in \mathcal{V}, \forall r \in \mathcal{R}$
7:      while $\exists r, U_r > \sum_{v\in\mathcal{V}} Q(v, r, F_v^T)$ and $\mathcal{B} \neq \Phi$ do
8:          Pick the largest $b_{v,r}$ in the sorted order of $\mathcal{B}$
9:          ${f'}_{v,r}^T \leftarrow 1$
10:         if $U_r - \sum_{k\in\mathcal{V}} Q(k, r, F_v^T) \geq Q(v, r, {F'}_v^T)$ then
11:             $F_v^T \leftarrow {F'}_v^T$
12:             Update $b_{v,l}, \forall l \in \mathcal{R}$
13:             Re-sort $\mathcal{B}$ in descending order
14:         end if
15:         Remove $b_{v,r}$ from $\mathcal{B}$
16:     end while
17: end procedure
```

The complexity of Algorithm 2 depends on ranking $\mathcal{B}$ and maintaining the rank in lines 7–16. The time complexity is thus $O(|\mathcal{V}||\mathcal{R}| \log |\mathcal{V}||\mathcal{R}|) \sim O(|\mathcal{V}| \log |\mathcal{V}|)$, since $|\mathcal{R}| \ll |\mathcal{V}|$. The algorithm is efficient enough when the candidate video set is small (*e.g.*, the most recently published videos are to be deployed) and the deployment is not adjusted very frequently – we will evaluate the performance of the deployment by varying the execution interval in Sec. 5.4.

## 5.4 Evaluating the Video Service Performance

### *5.4.1 Setup of Trace-driven PlanetLab Experiments*

▷ *Geo-distributed servers*. We implement our algorithms in C++ and deploy them on PlanetLab [29]. We use 5 PlanetLab nodes in China and 1 node in the US, corresponding to the regions studied. Each PlanetLab node acts as a video streaming server. This setting is only limited by the traces, and our design can scale when more regions are used in the prediction.

▷ *Video deployment and user requests*. We use the 1000 videos as described in the measurement studies in Sec. 3. The length of each video is 10 minutes and the streaming rate is 600 Kbps. Each server is assigned a number of videos in each time slot to serve according to the deployment algorithms. A PlanetLab node also generates the requests from viewers in that region, which are sent to different regions according to the video service scheme discussed in Sec. 5.

▷ *Traces and parameters*. We emulate the actual video visits recorded in our Youku traces in each time slot and the geographic distribution of viewers from our Weibo traces. The video requests in each time slot are uniformly distributed in a time slot. The upload bandwidth prices in the regions are randomly assigned in the range of $[0.5, 1.5]$ per MB, and the default budget for the deployment is $20,000$. Parameter $M$ is set as the same used in the prediction model (*i.e.*, 7 for old videos and $n$ for new videos published $n$ days ago), and $L$ is set to 6 so that the allocation is performed everyday.

▷ *Benchmark*. For benchmarking the performance, we measure the delay between the time a viewer issues a request and the time it receives the first $1,000$ KB of the video stream, which is the typical size of the video data before a viewer starts the playback. This delay consists mainly of the RTT between the request and service regions, as well as the queueing and processing delay at the server.

### *5.4.2 Experimental Results*

**An Overview of Performance Comparison**. First, we compare the performance of different upload bandwidth reservation schemes: (1) our scheme given in Algorithm 1; (2) a random allocation scheme in which a small amount of upload bandwidth is progressively allocated to a randomly selected region, until the total expense exceeds the budget; (3) a proportional allocation scheme where each time a small amount of upload bandwidth is allocated to a region selected randomly according to a probability that is proportional to the number of views in that region, *i.e.*, more upload bandwidth is allocated to a region with more viewing requests. Under each upload bandwidth reservation scheme, the same video deployment algorithm (Algorithm 2) is employed.

In Fig. 9, we observe that our bandwidth reservation scheme can achieve much smaller delays for the viewers, given that not only the number of viewing requests in each region, but also the user preference of different regions to receive videos from are incorporated in our design.

We also study the request load of these service regions under different upload reservation schemes. In Fig. 10, each sample represents the fraction of requests served by a region versus the region's rank. We observe that in our design, the request load is neither very uniformly distributed as achieved by the random scheme, nor highly skewed as in the proportional approach. Instead, our upload bandwidth allocation scheme supplies adequate bandwidth in each region that matches the number of requests in that region. The reason is that our reservation is based on the microblogging prediction framework, which provides a better estimation of future requests in a region than the proportional scheme.

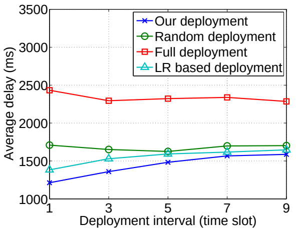


Fig. 12. Comparison under different deployment intervals.

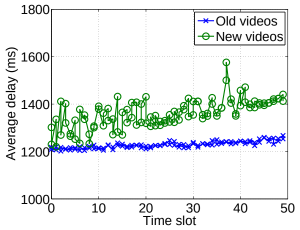


Fig. 13. Comparison between old and new videos.

**Impact of Deployment Interval**. The video deployment algorithm is carried out periodically, but frequent changes of deployment may face potential challenges: (1) videos may need to be frequently replicated among regions, incurring migration bandwidth consumption; (2) input features to the learning frameworks need to be collected frequently, incurring heavy load on the online microblogging system. We evaluate performance of the video service system when the video deployment is carried out in different intervals. Fig. 12 shows that for both our video deployment algorithm and the linear regression based scheme, the delay increases when the interval for deployment becomes larger; while for both the full replication and random deployment schemes, the delays remain at the same level regardless of the interval lengths. The results indicate that the deployment interval affects the performance of our design, and frequent adjustments of video deployment should be applied as long as the features from the online microblogging system can be collected timely.

**Impact of Deployment Budget**. We next compare our video deployment algorithm in Algorithm 2 with the following strategies: (1) a linear regression (LR) based scheme, *i.e.*, the same video service deployment algorithm but using the linear regression approach to predict the number and geographic distribution of video views; (2) a random deployment scheme, where each video is replicated in 1 randomly selected region in each time slot; (3) a full deployment scheme, where each video is replicated to all the 6 regions. As for upload bandwidth reservation, we use the same scheme as in Algorithm 1 but under different budgets. In Fig. 11, we observe that our video deployment algorithm performs much better than the other schemes, especially when the budget is small. When the budget is larger than $25,000$, which is about enough for purchasing bandwidths to serve all requests locally, all the schemes achieve similar delays. The observation indicates that compared to other schemes, our algorithm works effectively for a larger range of deployment budgets, since it more precisely deploy videos to a region with enough bandwidth, where they will be requested by many users in that region; while in the other deployment schemes, many viewing requests are redirected to regions without enough bandwidth capacity, resulting in large delays.

**Impact of Prediction Accuracy**. We compare the delays at viewers of old videos and new videos, respectively, which are treated differently in the prediction frameworks. Fig. 13 shows that viewers of old videos experience smaller delays than those of new videos. This is consistent with our evaluation results for the prediction accuracy in Sec. 4.5 — better deployment performance can be achieved with videos having a larger feature

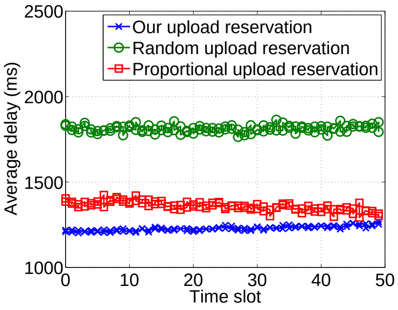


Fig. 9. Comparison under different upload bandwidth reservation schemes.

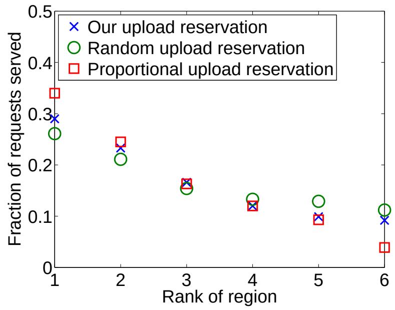


Fig. 10. Request load in different regions.

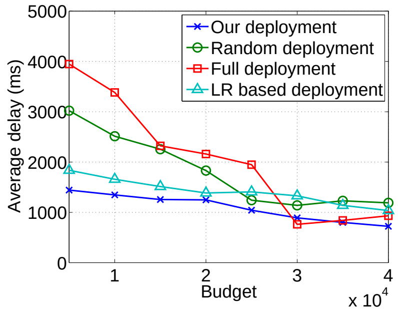


Fig. 11. Comparison under different video deployment schemes.

window, where the prediction accuracy is higher. As the prediction accuracy directly determines the video service deployment performance in our design, exploring new features from the microblogging services to improve the prediction accuracy is an approach to enhancing the streaming quality for video sharing sites.

## 6 Concluding Remarks

In this paper, we explore the connections between information propagation in a microblogging system and the number and distribution of actual views in a video sharing site, using extensive traces from two large-scale real-world microblogging and video sharing systems. Based on our discoveries of the connection between Weibo and Youku, we develop two neural network models for predicting the future number of video views and geographic distribution of the viewers, respectively. We further exploit the prediction frameworks to improve global-scale video service deployment. Our PlanetLab-based experiments illustrate the effectiveness of our deployment algorithms in reducing video access delays experienced by users, as compared to classical approaches without microblogging-assistant predictions.

## References

[1] H. Kwak, C. Lee, H. Park, and S. Moon, “What Is Twitter, a Social Network or a News Media?” in *Proc. of ACM WWW*, 2010.

[2] K. Lai and D. Wang, “Towards understanding the external links of video sharing sites: measurement and analysis,” in *Proc. of ACM NOSSDAV*, 2010.

[3] M. Cha, A. Mislove, and K. Gummadi, “A Measurement-Driven Analysis of Information Propagation in the Flickr Social Network,” in *Proc. of ACM WWW*, 2009.

[4] Z. Wang, L. Sun, X. Chen, W. Zhu, J. Liu, M. Chen, and S. Yang, “Propagation-based Social-aware Replication for Social Video Contents,” in *Proc. of ACM Multimedia*, 2012.

[5] YouTube, http://www.youtube.com/yt/press/statistics.html.

[6] V. Adhikari, S. Jain, Y. Chen, and Z. Zhang, “Reverse Engineering the YouTube Video Delivery Cloud,” in *Proc. of IEEE Hot Topics in Media Delivery Workshop*, 2011.

[7] Z. Wang, L. Sun, C. Wu, and S. Yang, “Guiding internet-scale video service deployment using microblog-based prediction,” in *Proc. of IEEE INFOCOM Mini-Conference*, 2012.

[8] G. Peng, “CDN: Content Distribution Network,” *arXiv preprint cs/0411069*, 2004.

[9] Y. Liu, Y. Guo, and C. Liang, “A survey on peer-to-peer video streaming systems,” *Peer-to-peer Networking and Applications*, vol. 1, no. 1, pp. 18–28, 2008.

[10] D. Xu, S. Kulkarni, C. Rosenberg, and H. Chai, “Analysis of a CDN–P2P Hybrid Architecture for Cost-Effective Streaming Media Distribution,” *Multimedia Systems*, vol. 11, no. 4, pp. 383–399, 2006.

[11] G. Marchuk, *Numerical Methods and Applications*. CRC, 1994.

[12] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, “I Tube, You Tube, Everybody Tubes: Analyzing the World’s Largest User Generated Content Video System,” in *Proc. of ACM SIGCOMM*, 2007, pp. 1–14.

[13] H. Li, H. Wang, and J. Liu, “Video Sharing in Online Social Network: Measurement and Analysis,” in *Proc. of ACM NOSSDAV*, 2012.

[14] M. Saxena, U. Sharan, and S. Fahmy, “Analyzing Video Services in Web 2.0: a Global Perspective,” in *Proc. of ACM NOSSDAV*, 2008.

[15] M. A. Balachander Krishnamurthy, Phillipa Gill, “A Few Chirps About Twitter,” in *Proc. of ACM WOSN*, 2008.

[16] N. Savage, “Twitter As Medium and Message,” *Communications of the ACM*, vol. 54, no. 3, pp. 18–20, 2011.

[17] J. Ritterman, M. Osborne, and E. Klein, “Using Prediction Markets and Twitter to Predict a Swine Flu Pandemic,” in *1st International Workshop on Mining Social Media*, 2009.

[18] R. Myers, D. Montgomery, G. Vining, and T. Robinson, *Generalized Linear Models*. Wiley, 2010.

[19] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[20] J. Yang and S. Counts, “Predicting the Speed, Scale, and Range of Information Diffusion in Twitter,” in *International AAAI Conference on Weblogs and Social Media*, 2010.

[21] G. Szabo and B. Huberman, “Predicting the Popularity of Online Content,” *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, 2010.

[22] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson Correlation Coefficient,” in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[23] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, “Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences With Livesky,” in *Proc. of ACM Multimedia*, 2009.

[24] E. Azoff, *Neural Network Time Series Forecasting of Financial Markets*. John Wiley & Sons, Inc., 1994.

[25] S. Chen, S. Billings, and P. Grant, “Non-Linear System Identification Using Neural Networks,” *International Journal of Control*, vol. 51, no. 6, pp. 1191–1214, 1990.

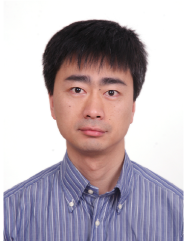
[26] K. Hornik, M. Stinchcombe, and H. White, “Multilayer Feedforward Networks Are Universal Approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[27] D. Specht, “A General Regression Neural Network,” *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.

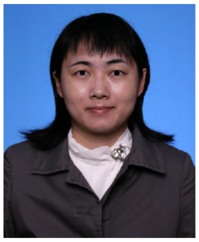
[28] R. Krishnan, H. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, “Moving Beyond End-to-End Path Information to Optimize CDN Performance,” in *Proc. of ACM IMC*, 2009.

[29] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, “Planetlab: an Overlay Testbed for Broad-Coverage Services,” *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3–12, 2003.

**Zhi Wang** received his B.E. and Ph.D. degrees in Computer Science in 2008 and 2014, from Tsinghua University, Beijing, China. He is currently an assistant professor in the Graduate School at Shenzhen, Tsinghua University. His research areas include online social network, mobile cloud computing and large-scale multimedia systems. He is a member of IEEE.



**Lifeng Sun** received his B.S. and Ph.D. degrees in System Engineering in 1995 and 2000 from National University of Defense Technology, Changsha, Hunan, China. He was an postdoctoral fellow from 2001 to 2003, an assistant professor from 2003 to 2007, an associate professor from 2007 to 2013, and currently a professor all in the Department of Computer Science and Technology at Tsinghua University. His research interests lie in the areas of online social network, video streaming, interactive multi-view video, and distributed video coding. He is a member of IEEE and ACM.



**Chuan Wu** received her B.E. and M.E. degrees in 2000 and 2002 from Department of Computer Science and Technology, Tsinghua University, China, and her Ph.D. degree in 2008 from the Department of Electrical and Computer Engineering, University of Toronto, Canada. She is currently an assistant professor in the Department of Computer Science, the University of Hong Kong, China. Her research interests include cloud computing, peer-to-peer networks and online/mobile social network. She is a member of IEEE and ACM.

**Shiqiang Yang** received the B.E. and M.E. degrees in Computer Science from Tsinghua University, Beijing, China in 1977 and 1983, respectively. From 1980 to 1992, he worked as an assistant professor at Tsinghua University. He served as the associate professor from 1994 to 1999 and then as the professor since 1999. From 1994 to 2011, he worked as the associate header of the Department of Computer Science and Technology at Tsinghua University. He is currently the President of Multimedia Committee of China Computer Federation, Beijing, China and the co-director of Microsoft-Tsinghua Multimedia Joint Lab, Tsinghua University, Beijing, China. His research interests mainly include multimedia procession, media streaming and online social network. He is a senior member of IEEE.