2022-07-19

Load-Adjusted Prediction for Proactive Resource Management and Video Server Demand Profiling

Obinna Izima
Pure Storage, oizima@purestorage.com

Ruairí de Fréin
Technological University Dublin, ruairi.defrein@tudublin.ie

Recommended Citation

O. Izima and R. De Fréin, "Load-Adjusted Prediction for Proactive Resource Management and Video Server Demand Profiling," 2022 33rd Irish Signals and Systems Conference (ISSC), Cork, Ireland, 2022, pp. 1-6, doi: 10.1109/ISSC55427.2022.9826200.

Funder: SFI

Load-Adjusted Prediction for Proactive Resource Management and Video Server Demand Profiling

Izima, Obinna and de Fréin, Ruairí
Ollscoil Teicneolaíochta Baile Átha Cliath,
Campas na Cathrach,
Ireland.
web: https://robustandscalable.wordpress.com

in: 2022 33rd Irish Signals and Systems Conference (ISSC). See also BIBTEX entry below.

BIBTEX:

```
@article{deFrein22Load,
  author={Izima, Obinna and de Fr\'{e}in, Ruair\'{i}},
  journal={2022 33rd Irish Signals and Systems Conference (ISSC)},
  title={Load-Adjusted Prediction for Proactive Resource Management
  and Video Server Demand Profiling},
  year={2022},
  volume={},
  number={},
  pages={1-6},
  doi={10.1109/ISSC55427.2022.9826200},
  url={https://ieeexplore.ieee.org/document/9826200},
  }
```

Load-Adjusted Prediction for Proactive Resource Management and Video Server Demand Profiling

Obinna Izima[†], Ruairí de Fréin[‡]
Pure Storage, Inc. Dublin, Ireland[†], Technological University Dublin, Ireland[‡]
Email: [†]oizima@purestorage.com, [‡]ruairi.defrein@tudublin.ie

***Abstract*—To lower costs associated with providing cloud resources, a network manager would like to estimate how busy the servers will be in the near future. This is a necessary input in deciding whether to scale up or down computing requirements. We formulate the problem of estimating cloud computational requirements as an integrated framework comprising of a learning and an action stage. In the learning stage, we use Machine Learning (ML) models to predict the video Quality of Delivery (QoD) metric for cloud-hosted servers and use the knowledge gained from the process to make resource management decisions during the action stage. We train the ML model weights conditional on the system load. Numerical results demonstrate performance gains of $\approx 59\%$ of the proposed technique over state-of-art methods. This gain is achieved using less computational resources.**

***Index Terms*—Machine Learning, Load-Adjusted Learning, Server Load Prediction, Quality-of-Delivery, Video Quality, Resource Management.**

I. Introduction

Advances in cloud computing and virtualization technologies have enabled the deployment of large-scale data centers which run a large number of Internet applications. With increasing Internet traffic, fuelled by video traffic [1], resource systems management and monitoring have become more complex, making load balancing and sharing even more critical [2]. As user demands increase or to avoid service quality degradation, additional compute resources are scaled up to satisfy Quality-of-Service (QoS) and Service Level Agreements (SLAs) [3], [4]. For instance, having the ability to accurately predict the video quality from a networked server cluster which is streaming to a client can help detect when there is a need to provision more resources, stabilize current network state or minimize resource usage based on a drop or increase in video quality. As a result, an intelligent and efficient resource management strategy is required to reduce the resource wastage, while ensuring sufficient performance are provided to cloud customers [5], [6]. Machine Learning (ML) regression-based models have been applied in predicting the video quality for a server cluster involved in a streaming session [4], [7]. The Load-Adjusted (LA) technique for predicting the quality of the streamed video was originally proposed in [4]. The LA approach is one in which the ML model weights are trained conditional on the load or the number of users currently accessing the video stream. The authors of [7] contributed an automatic parameterization of the regularization penalty through the use of the Elastic Net (EN) model which improved video quality predictions.

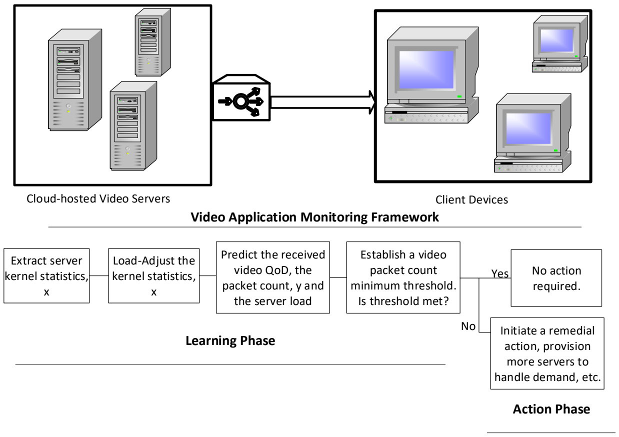
Cloud-hosted Video Servers
Client Devices
Video Application Monitoring Framework
Extract server kernel statistics, x
Load-Adjust the kernel statistics, x
Predict the received video QoD, the packet count, y and the server load
Establish a video packet count minimum threshold. Is threshold met?
Yes
No action required.
No
Initiate a remedial action, provision more servers to handle demand, etc.
Learning Phase
Action Phase

Fig. 1. The scope of the integrated framework is illustrated. By training ML regression models conditional on the load on the system, we demonstrate that we can predict the video packet count. The capability to realize real-time accurate QoD predictions can be used in a feedback loop to trigger a request for additional resources based on increased loads on the video server. The scope of this paper is limited to the learning phase.

Contribution: The first contribution of this paper is that we extend the scope of the LA technique to proactive resource management and video server demand profiling. Fig. 1 illustrates the purpose of our proposed learning framework. We demonstrate that by load-adjusting the streaming session data, that we can predict the video packet count, a Quality of Delivery (QoD) metric and use this as a reference to indicate when the network infrastructure may require additional computational resources to handle the increased load on the server [8]. This is because QoD measurements focus on the quality of the data delivery process and capture the the end-to-end performance of network services [9], [10]. In other words, these measurements can be used to infer how well the network and transport stack can deliver quality data. QoD, unlike QoS, is not service dependent. For video applications, QoD measures can be used to determine the ability to transmit video frames reliably [11]. This information can be useful for either knowing when to provision more server resources or when to reduce the server resources depending on the number of users involved in the streaming session. Given that we can

978-1-6654-5227-4/22/$31.00 2022 IEEE

accurately predict the packet count, we explore how the LA technique can be used in a feedback control system to detect or predict when there is a change in service demand, which we call a ”service change-point”. This service change-point could be the point or time interval when the system detects anomalies or increased requests for system resources based on a drop in the predicted video packet counts. We propose a framework which uses the network monitoring platform to initiate an alert when the video QoD falls below a specified threshold. The second contribution of this paper is a video server demand prediction technique using the LA approach. We show that by using a time-varying dataset in which the server demand profiles vary over time, that we can accurately predict the load. We demonstrate that using the LA technique, we can reliably predict the load on the video server with best Root Mean Squared Error (RMSE) and R-squared of ≈ 0.64 and 99% respectively. The ability to predict future server demand profiles could help a network manager to proactively manage the dynamicity of cloud provisioned resources.

This paper is organized as follows. In Section II we place our contributions in the context of the related literature. In Section III, we describe the network test-bed. In Section IV we describe the experimental setup. In Section V, we evaluate the efficacy of the learning approaches and present our results in Section VI. We present our conclusions in Section VII.

II. Related Works

The authors of [12] proposed a framework for dynamic application workload predictions which was used for making auto-scaling decisions for web resources. The authors demonstrated that by relying on historical access logs of web applications, they could predict the future workload trends from pre-computed workload patterns for a specific number of past time intervals. This information could be used to estimate future resource demands. Although the authors demonstrated the feasibility of their approach, this method may not be suitable for the type of applications we investigate in this paper given the specific QoD targets for video, like packet counts, in comparison with web traffic.

Saxena et al. in [13] proposed a framework for dynamic resource management in which future resource demands are estimated to ensure energy-efficient resource usage. The authors utilized Feed-forward Neural Network (FNN) models to predict future resource demands. They extended the study towards achieving auto-scaling of virtual resources based on a cluster of the estimates of the resource requirements. Related works in [14], [15] proposed host load estimation models based on a Recurrent Neural Network with a Long Short-Term Memory model (LSTM-RNN). However, due to the backpropagation algorithm utilized between recurrent layers, LSTM-RNN models incur long computational times despite their capacity to learn long-term dependencies and yield accurate models. The LA technique takes less time to train and is significantly computationally cheaper than these proposed methods. A key finding on the application of the LA technique for video quality predictions in [16], found that the LA model predictions were $\approx 50\%$ more accurate than existing baseline methods while only using 2% of the data. This is a significant improvement in accuracy in time and space complexities.

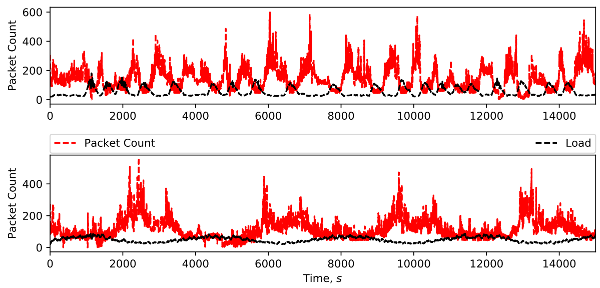

Fig. 2. Row 1 (R1): The video packet count, y, is illustrated for 14000s along with the number of active users accessing the video stream, K or TCPSCK for the flashcrowd-load trace. The video packet count and K for the periodic-load trace is shown. In both plots, as K increases, there is a drop in the video packet count received at the clients, which illustrates the dependence between the statistics.

The LA technique was first proposed in [4]. The authors proposed a generative model for QoD metrics prediction from the kernel-level metrics of a cloud-hosted video server. In LA learning, ML models are trained conditional on the load signal. In its simplest form, models are learned for each value of the load signal. This significantly speeds up the training time and is computationally cheap. In Un-Adjusted (UA) learning, ML models are trained regardless of the load value. This learning mechanism fails to capture the effect a user streaming session may have on another in a shared network resource. The study in [6] advances the results of the seminal work on LA learning proposed in [4]. Considering that time-varying loads in the system affect the estimation of QoD predictor parameters, the authors formulated the video quality prediction task as a supervised deconvolution problem. They used ideas from source separation to propose a LA version of UA video QoD prediction. The authors reported an improvement in the Signal-to-Noise-Ratio (SNR) for LA learning. They provided results from their evaluations using traces from the baseline UA approach [17] demonstrating that the LA learning was (1) faster, and (2) more accurate compared to the UA learning technique.

III. Network Test-Bed

Client machines access a Video-on-Demand streaming service delivered from a cloud-hosted server cluster in Fig. 1. A load generator dynamically distributes client video requests to the servers using either a periodic-load pattern or a flashcrowd-load pattern. The periodic-load patterns introduce clients following a Poisson process at an average rate of 30 per minute. This arrival rate is modulated by a sinusoidal function with a period of one hour and an amplitude of 20 clients. The flashcrowd-load pattern starts with a Poisson process where clients arrive at an average rate of 5 clients per minute, peaking at random events at a rate of 10 events per hour. Flash events see an increase in arrival rates to 50 clients per minute for about a minute and then gradually reduce to 5 clients per minute within 4 minutes. Device-level measurements, x, are collected on the servers using the Linux System Activity

Report (SAR). These statistics consist of operating system level metrics such as the total number of packets transmitted per second, the number of active processes, the TCP Socket count (TCPSCK). The TCPSCK feature of x can be used to estimate the load signal, which is the number of active clients viewing the video stream. In Fig. 2, we illustrate the dependence of the video packet count on the number of active users, the TCPSCK. As the load on the server increases, the TCPSCK increases and may result in a drop in the video QoD metric, the packet count. Using this video QoD metric, we propose a framework which determines if a sustained drop in the packet count is an indication of deteriorating network conditions. This information can then be used to proactively provision more server resources to resolve the situation leading to improved video QoD.

Problem Statement: Suppose we define a threshold video QoD value, y_{tr}, as a minimum allowed video packet count required to sustain the streaming session for a limited period of time. Our first objective is to predict the video packet count, y_i, using the features, x, given a time varying load, $K(i)$. In a follow-up approach, i.e. the action phase, our task will be to compare the y_i value with the y_{tr}, and initiate remedial actions if the former is below the latter. We limit the scope of this paper to the learning phase. A second objective is to predict the load on the video server using the features given a load-adjusted dataset. We generate time-varying datasets in which the load values vary over time. We will examine the LA and UA methods for video packet count and server load predictions using the XGBoost [18] and EN [19] algorithms. The XGBoost is an ensemble technique that utilizes the gradient boosting framework for ML predictions. The XGBoost algorithm employs second-order gradients and improved regularization to achieve more accurate approximations. Secondly, we apply the EN model as was done in [7] for preliminary predictions. The EN algorithm is a penalized LR model that incorporates the ℓ_1 as well as ℓ_2 penalties during training. A hyperparameter, α, is used in the EN algorithm for determining what weight each of the ℓ_1 and ℓ_2 penalties should receive. The α value ranges between 0 and 1. We have chosen the EN algorithm for its ability to automatically tune the hyperparameter and the XGBoost for its speed. We evaluate the performance of the resulting predictors using the Root Mean Square Error (RMSE) and the R-squared in percentage (%).

IV. Experimental Setup

LA Experimental Setup for Video QoD Prediction: The seminal work on the LA learning was proposed in [4]. The authors modelled the relationship between the device statistics, client video QoD metrics, and the system load using a linear model. According to the authors, the response of the server, with respect to kernel metric n, the n-th feature, to one request for video at time i is expressed as the sum of a load-based component $\hat{u}_i[n]$ (resources held by a user), and some deviation signal specific to a feature, $\epsilon_i[n]$,

$$x_i[n] = \hat{u}_i[n] + \varepsilon_i[n], \quad \text{where} \quad i \in \mathbb{Z}, x_i[n], \hat{u}_i[n] \in \mathbb{R}. \quad (1)$$

A feature is a metric at the operating system level, such as the TCPSCK. The feature set was constructed using SAR function, which provides system metrics for a given time period. In Eqn. 1, $x_i[n]$ refers to the n-th feature observed at time index i. The observed client QoD metric, y_i, is the packet count. The deviations from the expected performance are captured by the noise signal $\epsilon_i[n]$. The signal $\hat{u_i}[n]$ represents an increase in the server load for each user. For example, a request for additional resources θ_n made by the current client or a new client would initiate a feature response of the form:

$$x_i[n] = 2\hat{u_i}[n] + \varepsilon_i[n, 1] + \varepsilon_i[n, 2]. \quad (2)$$

The deviation from the ideal performance arising from the second user is denoted by $\epsilon_i[n, 2]$. Let us assume that at time i, the number of users requesting the service is $k[i]$. The response of the n-th feature to the time-varying load is

$$x_i[n] = \theta_n K(i) + \sum_{k=1}^{K(i)} \varepsilon_i[n, k]. \quad (3)$$

The load signal $\theta_n K(i)$ denotes the number of active users at time i times the resources one user uses, θ_n. The TCPSCK or load signal is $K[i]$. Training a Linear Regression (LR) model with the LA approach implies that the LR models are load-adjusted by training weights for each value of the load signal

$$\hat{y_i} \Big|_{K(i)=k} = \sum_{n=1}^{N} x_i[n]\beta[n] \Big|_{K(i)=k}. \quad (4)$$

To put it in a more general context, $y = f(x)$, where $f()$ is a ML algorithm such as Extreme Gradient Boosting (XGBoost) or Elastic Net (EN).

UA Experimental Setup for Video QoD Prediction: The model is not load-adjusted when all samples, regardless of load value, are used for training. A consequence of this is that the baseline UA approach for predicting client video QoD metrics from device statistics does not model the effect of the time-varying load. They assume that $K[i]$ is a constant.

LA Experimental Setup for Video Server Load Prediction: We employ the same methodology used for predicting the video QoD in these experiments. We outline the process of our evaluation in Procedure 1. The process begins by taking in the feature set, x as input in Line 1. Line 2 indicates that we set the target metric, $\hat{y}_i$, in Equation 4 as the TCPSCK. We generate time-varying load-adjusted datasets from the feature set, x in Line 3 based on the load ranges. We remove the TCPSCK and video QoD metrics from x in Line 4. We train the XGBoost model using the LA method and predict future load values in Lines 5–6. Finally, we compute the RMSE and R-squared metrics in Line 7.

Procedure 1 Process of Evaluation for Load Prediction

```
1: Input: $x$                    ▷ $x$ is the feature set comprising of the kernel metrics
2: $\hat{y}_i \leftarrow$ TCPSCK                               ▷ set the TCPSCK as $\hat{y}_i$
3: Generate time-varying feature set, $x$ based on the load value ranges
4: Remove the TCPSCK and video QoD metrics from $x$
5: Train the XGBoost model using the LA technique                  ▷ Equation 4
6: Predict future video server loads                             ▷ Testing data
7: Compute the RMSE and R-squared metrics
```

V. Numerical Evaluation

We compare the performance of the EN and XGBoost models using the LA learning approach with the UA technique, the baseline approach. We adopt the data preparation steps taken by the authors of [4]. There are 51043 observations with 297 features, and 15150 observations with 275 features for the periodic-load pattern and the flashcrowd-load pattern respectively. We begin by removing all non-numeric and constant value features from the data sets. We prepare the datasets for evaluation by adopting the validation set approach using a 60-40 split between the training and test data. The regularization technique in the EN model required a method for selecting the regularization parameter, λ, for the penalty function. To determine the value of the regularization parameter, λ, we applied 10-fold Cross-Validation (CV) for both the LA and UA learning approaches. The value obtained was used in subsequent learning and prediction experiments. The 10-fold CV was applied during the training and testing stages for the models. Different values for λ were determined for both the UA and LA algorithms. A sequence of values between 0.0001 and 1 was passed to the CV function to automate selection of the regularization parameter, λ. To train the XGBoost model, we selected a section of the algorithm's hyperparameters and configured the Scikit-learn GridSearchCV function to test each unique combination of hyperparameters and to record the error for each iteration. We then proceeded to train the LA and UA models using the parameters identified by the GridSearchCV function.

LA Learning: To evaluate the LA models, we extract the data from the feature set, x, for a range of K values, which are obtained from the TCPSCK value. For example, we generate a dataset for when the number of users in the system is within the range $30 < K < 39$ which is when there are 30 to 39 users assessing the VoD stream. Our goal in these LA experiments is to evaluate what happens when the number of active users suddenly doubles over a period of time. If we know that the system load has doubled, can we use this information to improve the video delivery system? We demonstrate what happens when the server load increases for the video delivery system shown in Fig. 1. In Fig. 3, the packet count recorded at the client device for 1000 seconds for server load values in the range $20 < K < 50$ abruptly transitioning to $50 < K < 100$ is shown. The vertical red line indicates the service change-point. We observe that the video QoD metric, the packet count drops when the number of active users doubles. This is due to the strain on the server resources in this shared environment. The mean packet count for $20 < K < 50$ users and $50 < K < 100$ is ≈ 163 packets/second and 81 packets/second respectively.

UA Learning: In our UA experiments, we adopt the approach described by the authors in [17]. We generate the train and test data using any sample from the data regardless of the load value. We ensure that the number of observations used to compare the UA models for a particular range of K values matches the number of samples used for the corresponding LA models. For instance, if there 1000 observations for a LA model for $10 < K < 20$, then we generate a dataset of 1000 observations for the UA model regardless of the K values. Using the validation set approach, we split the dataset in a 60-40 ratio for training and test purposes respectively.

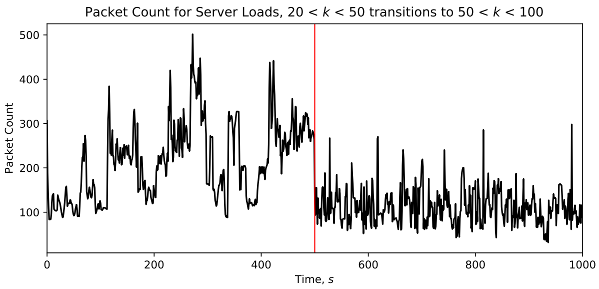

Fig. 3. The packet count delivered to the client for server load values in the range $20 < K < 50$ abruptly transitions to $50 < K < 100$ users. The vertical red line indicates the service change-point. There is drop in the packet count when the K value doubles. This drop in video QoD is illustrated.

TABLE I
The results of the LA-EN and UA-EN models for the 30 - 35, 60 - 65 user ranges, and the time-varying combined trace are shown. The LA-EN model offers better performance. (P) indicates that the dataset was drawn from the periodic-load trace. (P*) indicates that this is a combined time-varying data where the K values doubles mid-way through the range.

Dataset	LA-EN RMSE	LA-EN R^2	UA-EN RMSE	UA-EN R^2
$30 < K < 35$ (P)	26.70	84.40	30.51	70.10
$60 < K < 65$ (P)	23.94	83.29	36.47	73.16
$30 < K < 60$ (P*)	26.90	87.10	38.21	68.20

VI. Results

This section presents the results of the LA versus UA learning approaches for both video packet count and server load predictions. We report the evaluation results of the LA technique versus UA learning in different test scenarios, namely the flash-crowd and periodic-trace patterns. The LA models record superior performance compared with the UA models. In general, we adopt the naming convention of *"Learning Technique-ML Model"* in referring to the LA and UA ML models. For instance, LA-EN refers to the elastic net model learned via the load-adjusted technique.

To investigate the efficacy of utilizing the LA technique to make resource allocation and management decisions, we examine the performance of the LA-EN model for a time-varying dataset where the packet counts recorded for the 30 - 35 users transition to the 60 - 65 users range. The questions we hope to answer are: (1) Can we rely on the LA-EN automatic parameterization to gracefully handle the transition from a region where there is less demand for the system resources to a moment where this suddenly doubles? (2) Does the learning function adaptively respond to these changes especially around the time intervals leading to and following the increased demand for resources? Table I lists the performance of the LA-EN and UA-EN learning algorithms for the 30 - 35 users trace, 60 - 65 users trace and the combined time-varying dataset of both traces. (P) indicates that the dataset was drawn from the periodic-load trace. (P*) indicates that this is a combined time-varying data where the K values doubles mid-way through the

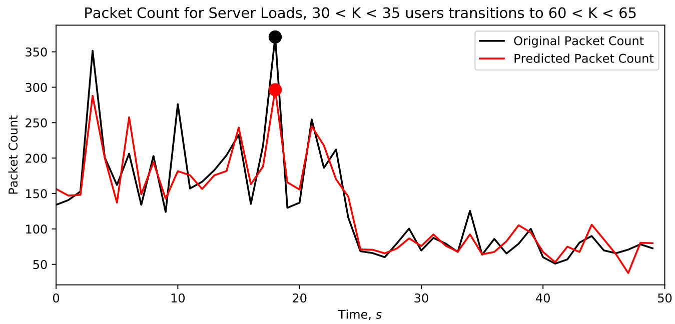

Fig. 4. The LA-EN model accurately predicts the packet counts for the time-varying dataset with load values in the range $30 < K < 35$ which doubles to $60 < K < 65$. The packet count predictions are plotted against the true packet counts. The LA-EN model predictions accurately approximates the original recorded packet counts.

TABLE II
RMSE, R SQUARED FOR THE LA-XGBOOST AND UA-XGBOOST MODEL PREDICTIONS FOR THE VIDEO PACKET COUNT GIVEN A RANGE OF K VALUES ARE LISTED. THE LA MODELS OFFER BETTER PREDICTIONS AND IS ABLE TO CAPTURE THE TRANSITION POINTS ACCURATELY. (P) AND (F) INDICATE THAT THE DATA WAS DRAWN FROM THE PERIODIC-LOAD AND FLASH-CROWD TRACES RESPECTIVELY. (P→F*) REFERS TO TIME-VARYING DATA TAKEN FROM P AND F TRACES.

	LA-XGBoost		UA-XGBoost	
Dataset	RMSE	R^2	RMSE	R^2
$30 < K < 35$ (F)	10.79	84.94	21.12	73.34
$30 < K < 39$ (P)	0.94	99.97	28.34	81.34
$60 < K < 70$ (P)	1.01	99.96	6.13	88.80
$30 < K < 60$ (P*)	23.47	90.15	24.04	89.67
$20 < K < 100$ (F*)	19.77	95.05	18.87	95.50
$20 < K < 100$ (P→F*)	24.20	81.86	28.67	78.82

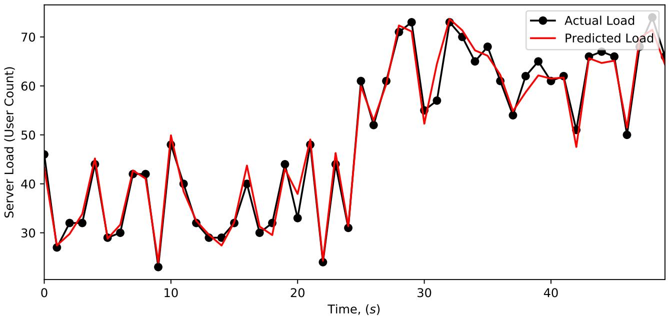

Fig. 5. The accuracy of the LA-XGBoost load predictions for the worst case scenario of RMSE and R-squared values of 3.13 and 97.74% are illustrated. The model accurately predicts the load on the server.

range. The packet counts for the time-varying trace lies in the range 51 to 370 packets/s. The LA-EN predictions generated for the dataset lie in the range 38 to 296 packets/s. The R-squared score for the data, which is a measure of the fit of the LA-EN regression model to the observed time-varying trace is 87.10%. Both metrics demonstrate the accuracy of the LA-EN model in adaptively learning the time-varying change in resource demands. In Fig. 4, we illustrate the accuracy of the LA-EN model predictions for the time-varying combined trace to understand what a RMSE of 26.93 and an R-Squared score of 87.10% means. The original packet count recorded for the time-varying trace is compared with the LA-EN model predicted values. The mean true packet count when there is less demand for the system resources, i.e., the 30 - 35 users region is 191 packets/s. The mean true packet count when the demand increases to double the amount, i.e., the 60 - 65 users region is 133 packets/s. The mean LA-EN packet count predictions are 187 packets/s and 131 packets/s for the 30 - 35 users region and 60 - 65 users regions respectively. The main point we observe is that the LA-EN predictions are much better during increased levels of demand. This accuracy in predictions is crucial as a misprediction would impact SLAs that guarantee the expected video QoD. This result reinforces one of the key findings from the study in [16]. The authors had reported that the video QoD drops for $K \geq 30$. They had shown that the variance in the video QoD increase up to when $K = 30$ and drops thereafter. The prediction error would naturally tend to zero when there is no variance. In Fig. 4, we observe a spike for the original packet count (represented by the circle) at timestamp 18s; the LA-EN model accurately tries to approximate this peak (represented by the red circle). The original packet count value at this point is 370 packets and the LA-EN predicts 296 packets. The prediction is off by 74 packets.

Table II lists the results of the XGBoost algorithm using the LA and UA techniques. (P) and (F) indicate that the data was drawn from the periodic-load and flash-crowd traces respectively. (P→F*) refers to time-varying data taken from P and F traces. For all cases considered, the LA-XGBoost model outperforms the UA-XGBoost in all but one case. The UA-XGBoost model produces a better RMSE and R-squared metrics of 18.87 and 95.50% respectively. This is ≈ 0.9 and 0.45% improvements over the LA-XGBoost RMSE and R-squared vales. However, in every other scenario investigated, the LA-XGBoost model produces a better performance over the UA-XGBoost. The LA-XGBoost algorithm produces its best performance with the $30 < K < 39$ using the periodic-load dataset by recording RMSE and R-squared values of 0.94 and 99.97% respectively. Conversely, for the same dataset, the UA-XGBoost model achieves RMSE and R-squared values of 28.34 and 81.34% respectively. The LA-XGBoost achieves a performance gain of 27.40 and 18.56% in terms of RMSE and R-squared over the UA-XGBoost.

Video Server Load Prediction: We present the results from video server load predictions. Table III lists the results of the server predictions for K using the LA-XGBoost model. (P) and (F) indicate that the data was drawn from the periodic-load and flash-crowd traces respectively. (P*) and (F*) refer to time-varying data in which the load values doubles mid-way. (P→F*) refers to time-varying data taken from P and F traces. We demonstrate that the model achieves accurate predictions of the load on the server. The model achieves RMSE and R-squared metrics of 3.13 and 97.74 respectively for the worst case K, number of active users prediction. We demonstrate this accuracy with the plot in Fig. 5. This dataset starts with

TABLE III
RMSE, R SQUARED FOR THE LA-XGBOOST PREDICTIONS FOR THE SERVER LOAD, K IS LISTED. THE LA-XGBOOST MODEL ACHIEVES ACCURATE PREDICTIONS OF THE SERVER LOADS WITH RMSE AND R^2 VALUES OF 2.80 - 0.64 AND 97.74 - 99.98 RESPECTIVELY. (P) AND (F) INDICATE THAT THE DATA WAS DRAWN FROM THE PERIODIC-LOAD AND FLASH-CROWD TRACES RESPECTIVELY. (P*) AND (F*) REFER TO TIME-VARYING DATA IN WHICH THE LOAD VALUES DOUBLES MID-WAY. (P→F*) REFERS TO TIME-VARYING DATA TAKEN FROM P AND F TRACES.

		LA-XGBoost	
Dataset	Load	RMSE	R^2
$20 < K < 147$ (F*)	K	2.81	99.33
$20 < K < 45$ (P*)	K	2.75	98.19
$20 < K < 100$ (P)	K	3.13	97.74
$20 < K < 100$ (F→P*)	K	2.17	99.50
$20 < K < 60$ (F*)	K	0.67	99.32
$20 < K < 100$ (P→F*)	K	1.77	99.98

a low region where the server load is within the range $20 < K < 50$ and then transitions to $50 < K < 100$. The plot shows the accuracy of the LA model in capturing the increased number of users. This demonstrates the efficacy of accurately predicting the number of users or the load on the system. This is a crucial information that can be used for knowing when to initiating auto-scaling policies to handle the increased server demand. This results demonstrate that by training the XGBoost algorithm regression weights conditional on the load value in the system, we can achieve accurate predictions of the video server loads. The baseline approach, UA which trains the ML models without explicitly modelling the load in the system is unreliable even though it realized potentially accurate results. Based on the results of the empirical study in this study and backed up the work in [16], adopting a LA approach would yield more accurate predictions. However, in these types of scenarios, if the LA strategy is not used, the load on the system may negatively impact the predictions.

VII. Conclusion

We introduced a method for improving the level of video packet counts received by a client device by up to 8 packets/s. This performance gain was achieved by adopting the load-adjusted strategy. This improved performance gain using the XGBoost is ideal for detecting when a sustained drop in received packet counts could be a sign of a network problem. This result is significant owing to the fact that the XGBoost algorithm is designed to be both computationally efficient and effective. This can help a network manager know when to initiate a remedial action or provision addition resources to handle the increased requests for video resources. We also demonstrated suitability of the LA-XGBoost in predicting the server load. We demonstrated this by showing accuracies in predicting the number of active users and the total load on the system. The accuracies in R-squared are $\approx 97\%$ and above and RMSE metrics are below 3.13. We have also given evidence that the LA-EN automatic parametrization is able to realize accurate predictions of the received packet count for cases when the server demands increases.

The ability of the LA models to capture points of transition from low to high service utilization makes a case for integrating this model with an auto-scaling facility or a feedback loop as proposed in [20]. In the future, we plan to integrate the remaining part of our framework, namely the action phase, in order to apply the relevant knowledge gathered during the learning phase to various network decisions such as resource allocation and auto-scaling. The gains of integrating service elasticity with the LA technique can aid in preserving and maintaining the agreed service agreements.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/SIRG/3459.

References

[1] Cisco Systems 2018, “Cisco Visual Networking Index: 2017-2022 White Paper,” *Cisco*, 2018.

[2] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, “A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications,” *IEEE Access*, vol. 9, pp. 41 731–41 744, 2021.

[3] C. Sharma, P. K. Tiwari, and G. Agarwal, “An Empirical Study of Different Techniques for the Improvement of Quality of Service in Cloud Computing,” in *Data Eng. for Smart Systems*. Springer, 2022, pp. 333–340.

[4] R. de Fréin, “Effect of System Load on Video Service Metrics,” in *IEEE ISSC*, 2015, pp. 1–6.

[5] A. Shahidinejad, M. Ghobaei-Arani, and M. Masdari, “Resource Provisioning Using Workload Clustering in Cloud Computing Environment: A Hybrid Approach,” *Cluster Computing*, vol. 24, no. 1, pp. 319–342, 2021.

[6] R. de Fréin, “Source Separation Approach to Video Quality Prediction in Computer Networks,” *IEEE Communication Letters*, vol. 20, no. 7, pp. 1333–1336, 2016.

[7] O. Izima, R. de Fréin, and M. Davis, “Video Quality Prediction Under Time-Varying Loads,” in *IEEE CloudCom*, 2018, pp. 129–132.

[8] R. de Fréin, “Load-Adjusted Video Quality Prediction Methods for Missing Data,” in *IEEE ICITST*, 2015, pp. 314–319.

[9] T. N. Minhas, “Network Impact on Quality of Experience of Mobile Video,” Ph.D. dissertation, BTH, 2012.

[10] M. T. Vega, C. Perra, and A. Liotta, “Resilience of Video Streaming Services to Network Impairments,” *IEEE Trans. Broad.*, vol. 64, no. 2, pp. 220–234, 2018.

[11] O. Izima, R. de Fréin, and A. Malik, “A Survey of Machine Learning Techniques for Video Quality Prediction from Quality of Delivery Metrics,” *Electronics*, vol. 10, no. 22, p. 2851, 2021.

[12] W. Iqbal, A. Erradi, and A. Mahmood, “Dynamic Workload Patterns Prediction for Proactive Auto-scaling of Web Applications,” *J. of Net. and Comp. App.*, vol. 124, pp. 94–107, 2018.

[13] D. Saxena and A. K. Singh, “A Proactive Autoscaling and Energy-efficient VM Allocation Framework Using Online Multi-resource Neural Network for Cloud Data Center,” *Neurocomputing*, vol. 426, pp. 248–264, 2021.

[14] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, “Host Load Prediction with Long Short-Term Memory in Cloud Computing,” *J. of Supercomp.*, vol. 74, no. 12, pp. 6554–6568, 2018.

[15] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen, “Long Short Term Memory Recurrent Neural Network Based Multimodal Dimensional Emotion Recognition,” in *Intl. W. on Audio/Visual Emotion Challenge*, 2015, pp. 65–72.

[16] R. de Fréin, “Take off a load: Load-adjusted video quality prediction and measurement,” in *IEEE Inter. Conf. on Comp. and IT*, 2015, pp. 1886–1894.

[17] R. Yanggratoke, J. Ahmed, J. Ardelius, C. Flinta, A. Johnsson, D. Gillblad, and R. Stadler, “Predicting Service Metrics for Cluster-based Services using Real-time Analytics,” in *CNSM*, 2015, pp. 135–143.

[18] T. Chen and C. Guestrin, “XGBoost: Reliable Large-scale Tree Boosting System,” in *SIGKDD Conf. on KDDM, San Francisco, CA, USA*, 2015, pp. 13–17.

[19] H. Zou and T. Hastie, “Regularization and Variable Selection Via the Elastic-Net,” *J. Roy. Stat. Soc.*, vol. 67, no. 2, pp. 301–320, 2005.

[20] O. Izima, R. de Fréin, and A. Malik, “Codec-Aware Video Delivery Over SDNs,” in *IFIP/IEEE Intl. Symp. on Int. Net. Managt. (IM)*, 2021, pp. 732–733.