Xatu: Richer Neural Network Based Prediction for Video Streaming

Yun Seong Nam¹*, Jianfei Gao¹, Chandan Bothra¹, Ehab Ghabashneh¹, Sanjay Rao¹, Bruno Ribeiro¹, Jibin Zhan², Hui Zhang²

¹Purdue University, ²Conviva, *Google

This work was done when Yun Seong Nam was at Purdue University.
Internet video delivery ecosystem

- Internet video is delivered over:
  - Heterogeneous networks: WiFi, wired, 3G/4G LTE
  - Highly varying or challenging network conditions
Internet video streaming today

- Quality of experience (QoE) issues are common place.
- Many factors constitute QoE
  - Avoiding rebuffering
  - Ensuring as high a quality as possible

Low QoE adversely impacts user engagement and revenue
Background: Adaptive Bitrate Streaming

A video clip is encoded with multiple qualities (bitrates)
Background: Adaptive Bitrate Streaming

Video encoded at each bitrate is split into chunks
Background: Adaptive Bitrate Streaming
ABRs critically rely on predictions

4 sec of chunks in the player buffer
ABRs critically rely on predictions.
Contributions

- Expose limitations of existing approaches to predicting chunk download times.
  - Based on insights from video sessions of real users.
- **Xatu**, novel prediction approach based on a customised neural network.
- Evaluations showing Xatu’s promise:
  - 24% reduction in prediction error relative to state of the art. (CS2P, SIGCOMM 2016)
  - Integration with multiple ABRs with substantial performance improvement.
Existing prediction approaches

- chunk download request sent
- first byte received
- chunk download finished
- downloading

Video Client

Video Host
Existing prediction approaches

- Neglects TTFB (Time to First Byte).
- Assume chunk download times mainly depend on network throughput.
- Assume throughput independent of chunk size.
Existing prediction approaches

- State-of-the-art: CS2P [Sigcomm 2016]
  - Learns from prior video sessions.
  - Considers features such as ISP, CDN, access technology, and time of day.
  - Partitions video sessions based on these features, and uses a Hidden Markov Model for each combination of features.
What our data analysis reveals..

- **100K video sessions** from real users
  - Collected over three months in 2017 from a content publisher in US.
  - Sessions spread over **89 ISPs, 1406 cities**, and 2 CDNs.
What our data analysis reveals..

- **100K video sessions** from real users
  - Collected over three months in 2017 from a content publisher in US.
  - Sessions spread over 89 ISPs, 1406 cities, and 2 CDNs.

**CDF (Perc. of chunks)**

% of download time due to TTFB

**TTFB contributes more than 40% of download times for 20% of the chunks.**
What our data analysis reveals..

- **100K video sessions** from real users
  - Collected over three months in 2017 from a content publisher in US.
  - Sessions spread over 89 ISPs, 1406 cities, and 2 CDNs.

**Figure:**

- TTFB contributes more than 40% of download times for 20% of the chunks.
- Throughput tends to be higher for larger chunk size.
Does clustering improve prediction accuracy?

- **CS2P**: Per-cluster HMM; **Global-CS2P**: HMM on sessions across all data.

- **What our data shows:**
  - In about 35% of clusters, CS2P shows similar or even worse prediction error than Global-CS2P.
  - Using features such as ISP, CDN etc. not always helpful and can even hurt.

- **Why?**
  - Apriori clustering reduces data-set to learn from.
  - Assumes sessions in the partition have similar network performance: not always true!
Xatu: Motivation

- Model sequences with multiple chunk-dependent features, not just throughput.
- Learn from similar sessions without pre-partitioning.
Xatu: Custom Architecture

LSTM layer
**Conventional approach**
- Difficult to interpret which sessions are considered similar.

**Xatu’s custom approach**
- Gate mask helps in interpretability.
Temporal features of past ‘k’ chunks: $d_{t-k}^{(i)} \ldots d_t^{(i)}$: size, TTFB, download time, throughput.

Sequence modelled using LSTM to predict next value(s) in a time series.
Xatu Architecture - Static feature block

- Video session ‘j’ with ‘n’ static features.
- Static features: $s_n^{(i)}$
- Output: gate mask, $z^{(i)}$

**Embedding layer of static feature**

- FullyConnected Layer + Activation (Sigmoid)
- EmbeddingConcat
- $s_1^{(j)}$, $s_2^{(j)}$, $s_3^{(j)}$, ..., $s_n^{(j)}$

**Temporal features block**

- LSTM Output Activation (Tanh)
- LSTM Layer
- FullyConnected Layer + Activation (Tanh)
- TemporalConcat
- $d_{t-k}^{(j)}$, $d_{t-k-1}^{(j)}$, ..., $d_t^{(j)}$

**Static features (n)**: ISP, CDN, time of the day and city.
Xatu Architecture - Selective Gate

- Selective gate combines the static and temporal blocks.

**Static**

- Embedding layer of static feature
  - FullyConnected Layer + Activation (Sigmoid)
  - EmbeddingConcat
  - Static features ($n$): ISP, CDN, time of the day and city.

**Temporal with LSTM**

- FullyConnected Layer + Activation (Tanh)
- LSTM Layer
- Temporal features block
  - Temporal features: chunk download time, TTFB, size of chunk and network throughput
Xatu is interpretable

- Gate mask output from static block: $z^{(l)}$
- Using PCA\(^3\), project gate masks into 2D space.
- Closer dots indicate Xatu identifies corresponding sessions have similar performance.
Xatu is interpretable:

Sessions with same CDN tend to have similar performance
Xatu is interpretable:

Sessions with same CDN tend to have similar performance

Time of day also plays a noticeable role
Evaluation Methodology

● How effective is Xatu in achieving better prediction accuracies than CS2P?
● How do better predictions translate into better performance for video streaming algorithms?
  ○ Integrate Xatu with well known ABR algorithms.
Prediction accuracy - Xatu vs. CS2P

\( y_t \): Actual throughput,
\( \hat{y}_t \): Predicted throughput,
\( C^{(j)} \): # of chunks in video session, j.

Mean Normalised Absolute Error (NAE) per session:

\[
\frac{1}{C^{(j)}} \sum_{t=1}^{C^{(j)}} \left| \frac{y_t^{(j)} - \hat{y}_t^{(j)}}{y_t^{(j)}} \right|
\]
Prediction accuracy - Xatu vs. CS2P

\[ \text{Mean Normalised Absolute Error (NAE) per session:} \]

\[
\frac{1}{C^{(i)}} \sum_{t=1}^{C^{(i)}} \left| \frac{y_t^{(j)} - \hat{y}_t^{(j)}}{y_t^{(j)}} \right|
\]

\( y_t \): Actual throughput,
\( \hat{y}_t \): Predicted throughput,
\( C^{(i)} \): # of chunks in video session, \( j \).

Reduce median and 90\% ile of mean NAE by 23.8\% and 41.8\%
Does Xatu benefit ABR algorithms?

- Integrate Xatu with 2 representative ABR algorithms: MPC and FuguABR
  - MPC: Well studied algorithm based on Model Predictive Control.
  - FuguABR: Recent algorithm that uses a stochastic controller.
Does Xatu benefit ABR algorithms?

- Integrate Xatu with 2 representative ABR algorithms: **MPC** and **FuguABR**
  - MPC: Well studied algorithm based on Model Predictive Control.
  - FuguABR: Recent algorithm that uses a stochastic controller.

**FuguNN**

- Fully connected neural network.
- Predicts *probabilistic distribution* of download times.
- Only *temporal features* and does not model TTFB.

**FuguABR**

- ABR algorithm with stochastically optimal controller.
Does Xatu benefit ABR algorithms?

- Integrate Xatu with 2 representative ABR algorithms: MPC and FuguABR
  - MPC: Well studied algorithm based on Model Predictive Control.
  - FuguABR: Recent algorithm that uses a stochastic controller.

<table>
<thead>
<tr>
<th>FuguNN</th>
<th>XatuDist</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Fully connected neural network. *Predicts probabilistic distribution of download times *Only temporal features and does not model TTFB.</td>
<td>*Adding uncertainty quantification to Xatu to get Gaussian distribution of download times. *For fairness, disable static features and TTFB.</td>
</tr>
</tbody>
</table>

FuguABR: *ABR algorithm with stochastically optimal controller.
FuguABR + XatuDist v/s FuguABR + FuguNN

- QoE-SSIM (Linear combination of three metrics)
  - Average SSIM
  - Rebuffering Ratio
  - SSIM change magnitude

XatuDist observes higher QoE.
FuguABR + XatuDist v/s FuguABR + FuguNN

XatuDist achieves lower rebuffering ratio, median ~ 0 while FuguNN has median rebuffering of 2%.
Summary of other results:

- **Relative to Pensieve** (reinforcement learning approach), Xatu+MPC improves the median and 90%tile QoE by 29.2% and 5.8% respectively.

- Compared with **CS2P+MPC**, Xatu+MPC reduces the rebuffering events by 26% and improves the median average bitrate change magnitude by 17.4%.
Extensibility of Xatu to new information

- Generalize Xatu to other datasets and extend with new features.
- Collect a smaller data-set through controlled experiments which includes information about which CDN layer [Edge or Remote] each chunk is served from.
Extensibility of Xatu to new information

- Generalize Xatu to other datasets and extend with new features.
- Collect a smaller data-set through controlled experiments which includes information about which CDN layer [Edge or Remote] each chunk is served from.

Throughput depends on where a video chunk is served from.
Extensibility of Xatu to new information

- Generalize Xatu to other datasets and extend with new features.
- Collect a smaller data-set through controlled experiments which includes information about which CDN layer [Edge or Remote] each chunk is served from.

New feature (CDN layer) improves the median and 90%ile prediction error by 13.1% and 31.5%.
Conclusion

- Xatu achieves 24% reduction in prediction error relative to state of the art, CS2P, Sigcomm 2016.
- Xatu’s custom architecture helps in interpretability and reduces prediction error by 9.4%.
- Xatu integrates with multiple ABRs and achieves significantly better performance.
- Xatu is extensible and adding new features reduces prediction error by 13%.
- Dataset available at: https://github.com/Purdue-ISL/XatuDataset
Thanks!

Q & A