Hyperprofile-based Computation Offloading in Mobile Edge Networks

Andrew Crutcher[1], Caleb Koch[2], Kyle Coleman[3]
Mentors: Jon Patman[4], Prasad Calyam[4], Flavio Esposito[3]

October 22, 2017

[1]Southeast Missouri State University, [2]Cornell University, [3]Saint Louis University, [4]University of Missouri – Columbia

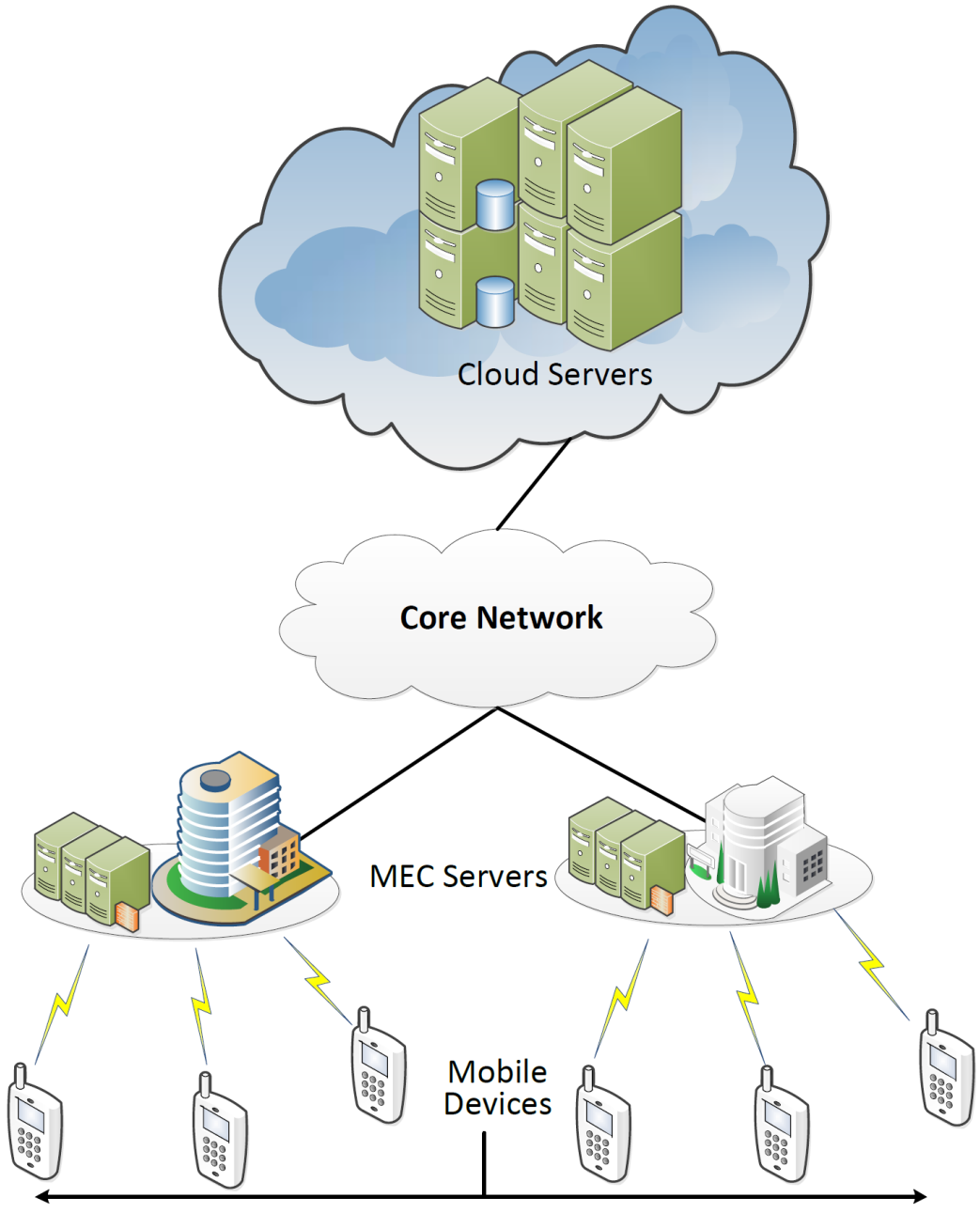
What we'll cover

- Background and problems
- General overview of our solution
- Distance metrics
- Developing our prediction model

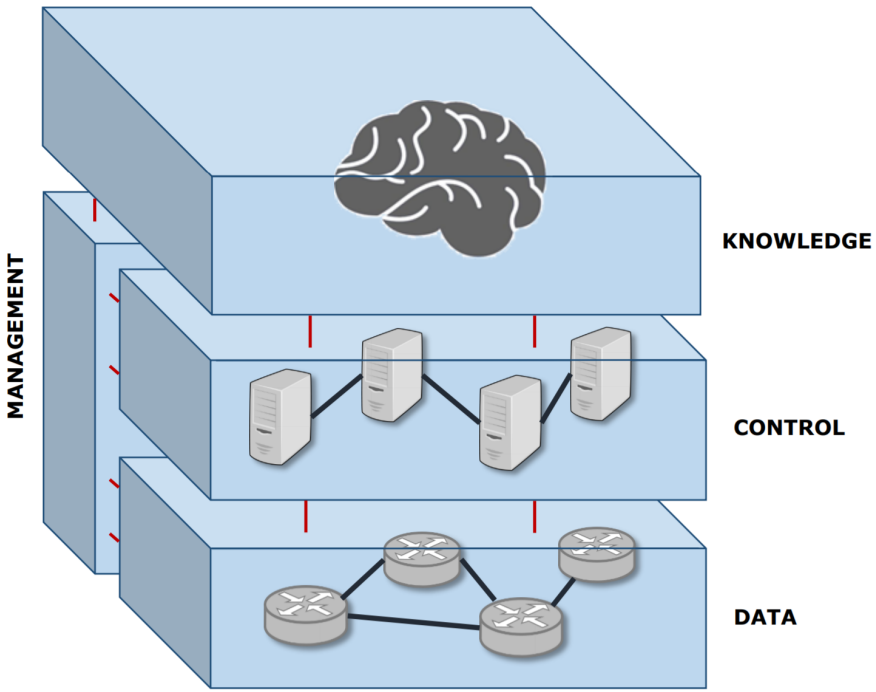
Background Information

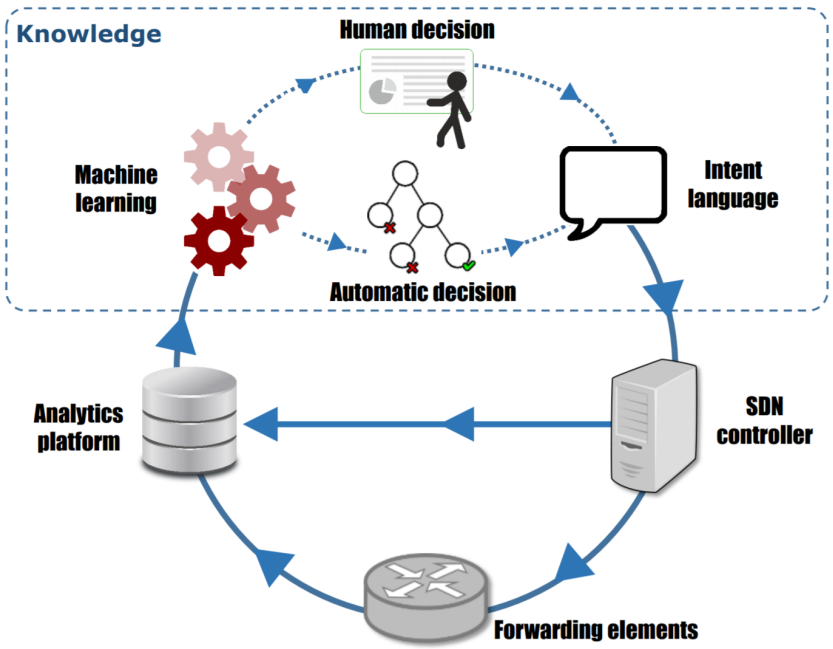
- Edge networks
- KDN / SDN
- Computation Offloading

Edge Network



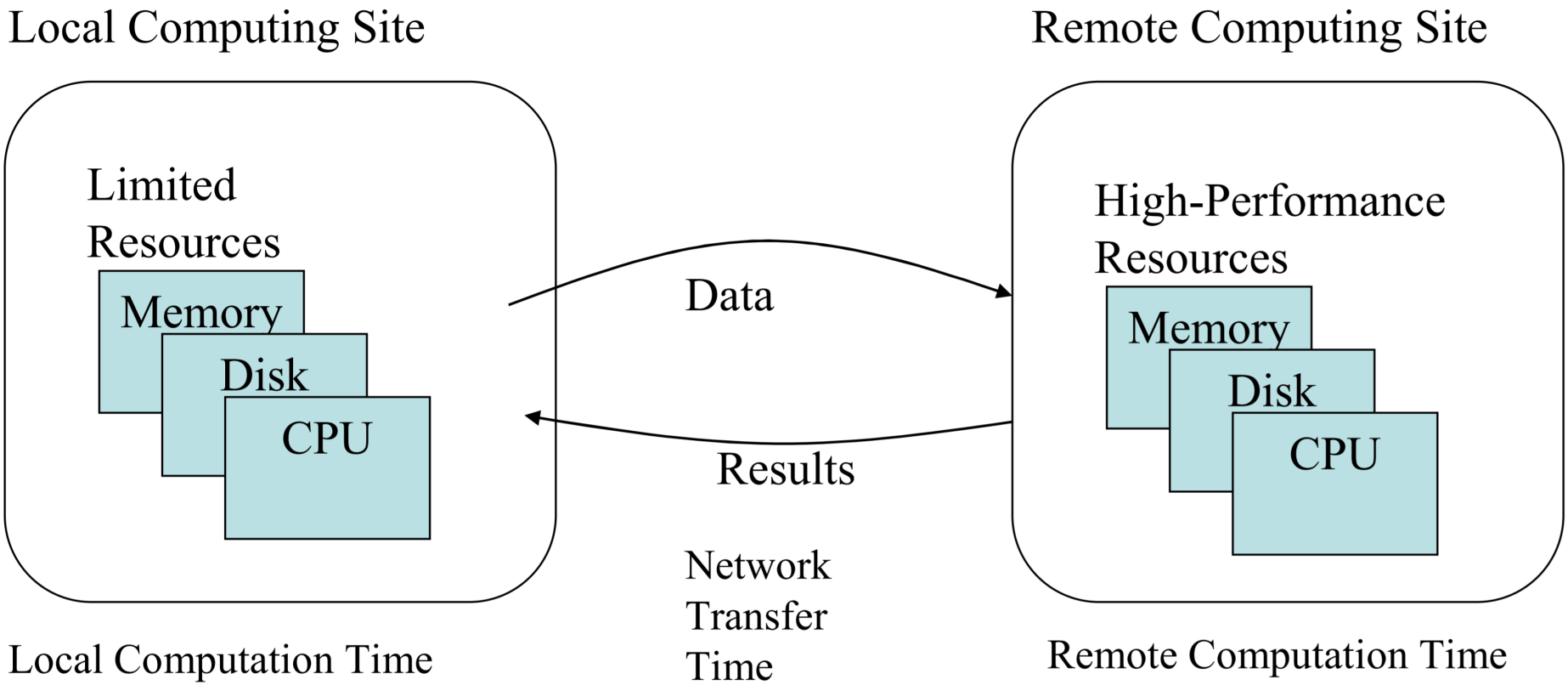
Knowledge Defined Networking





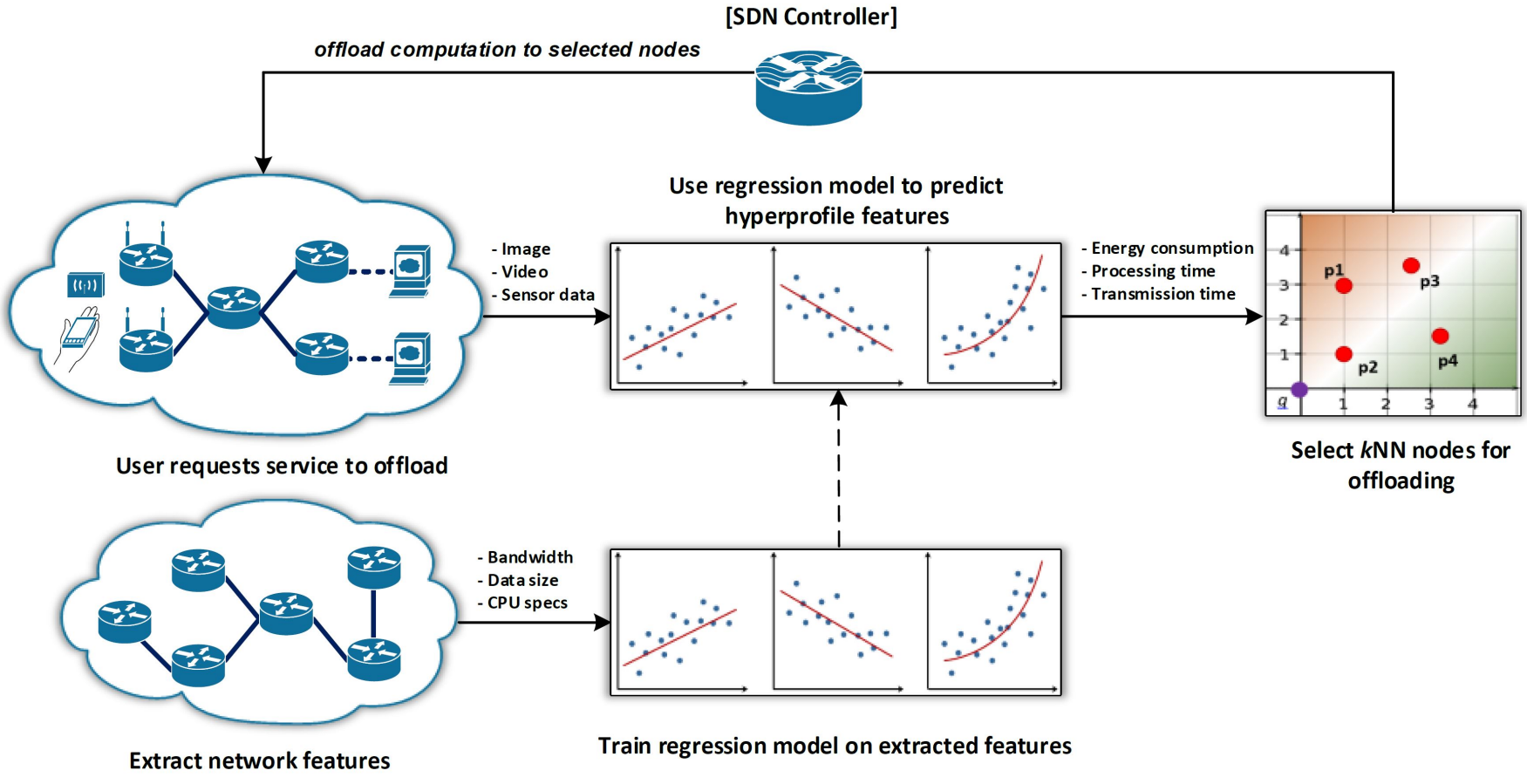
Mestres, Albert, et al. “Knowledge-Defined Networking.” arXiv preprint arXiv:1606.06222 (2016).

Computation Offloading



Wolski, Rich, et al. “Using bandwidth data to make computation offloading decisions.” Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on. IEEE, 2008.

Our Solution



Hyperprofile

Using our network model, we can build a hyperprofile

Definition

A hyperprofile for a set of edge server nodes is a collection of profiles that consist of dynamic metrics that are predicted in real-time by a predefined network model and related data sets.

The hyperprofile can help with our offloading decision

How can we query points in the hyperprofile?

The k-Nearest Neighbors algorithm suits the problem representation

Definition

Consider a set P of points and a point q then

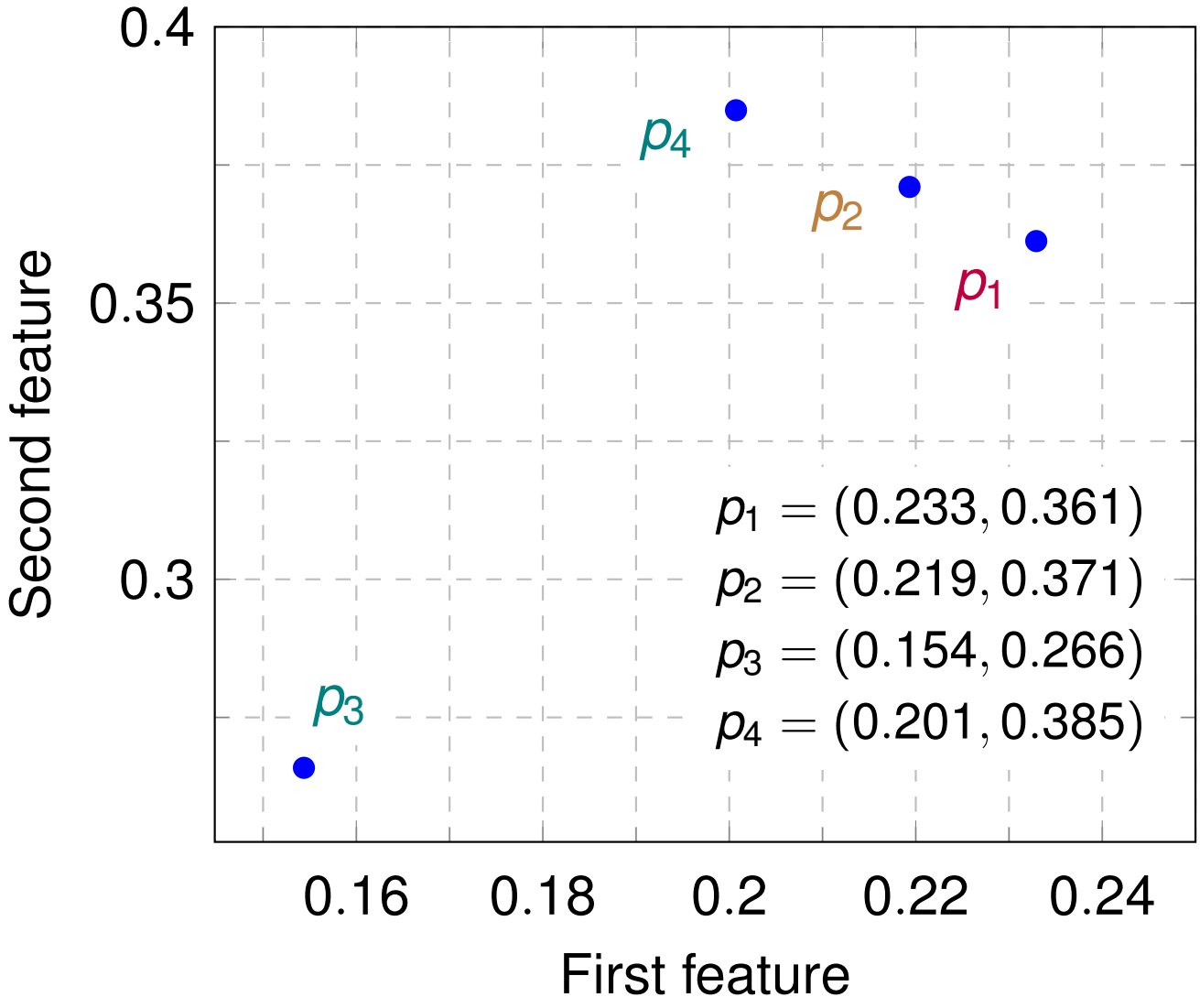
$$p \in k\text{NN}(q) \text{ iff } |\{j \in P : d(j,q) < d(p,q)\}| < k$$

where d is a distance metric (typically Euclidean).

Alternatives to Euclidean Distance

- kNN with a Euclidean distance metric minimizes the sum of squares of coordinates
  - e.g. if $x_1^2 + y_1^2 < x_2^2 + y_2^2$ then (x_1, y_1) is selected by kNN first
- Other methods e.g. in Chen 2015 use a minimization of the sum of the coordinates i.e. $x + y$ rather than $x^2 + y^2$
- So what's the difference?

Example scenario where $k = 3$ and $q = (0, 0)$



✓ Both ✓ Euclidean ✓ Rectilinear

What's happening when there's a mismatch?

Suppose $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$ and

$$x_1^2 + y_1^2 < x_2^2 + y_2^2 \tag{1}$$

$$x_2 + y_2 < x_1 + y_1 \tag{2}$$

then

$$\text{NN}(0,0) = \{p_1\}$$

$$\text{NN}_+(0,0) = \{p_2\}$$

where NN_+ returns the nearest neighbor based on a sum of coordinates.

Proposition

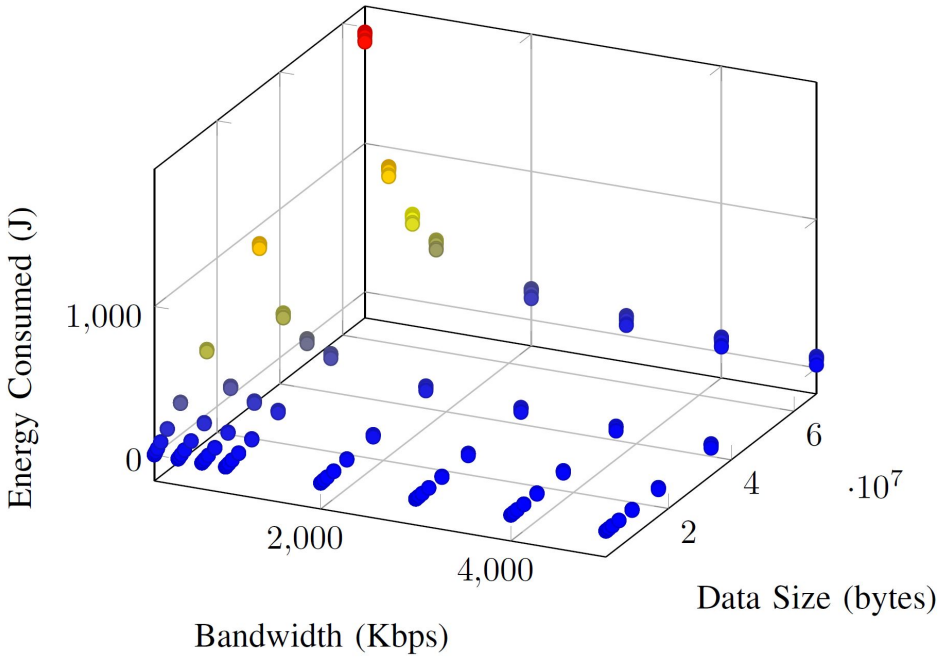
If (1) and (2) hold then

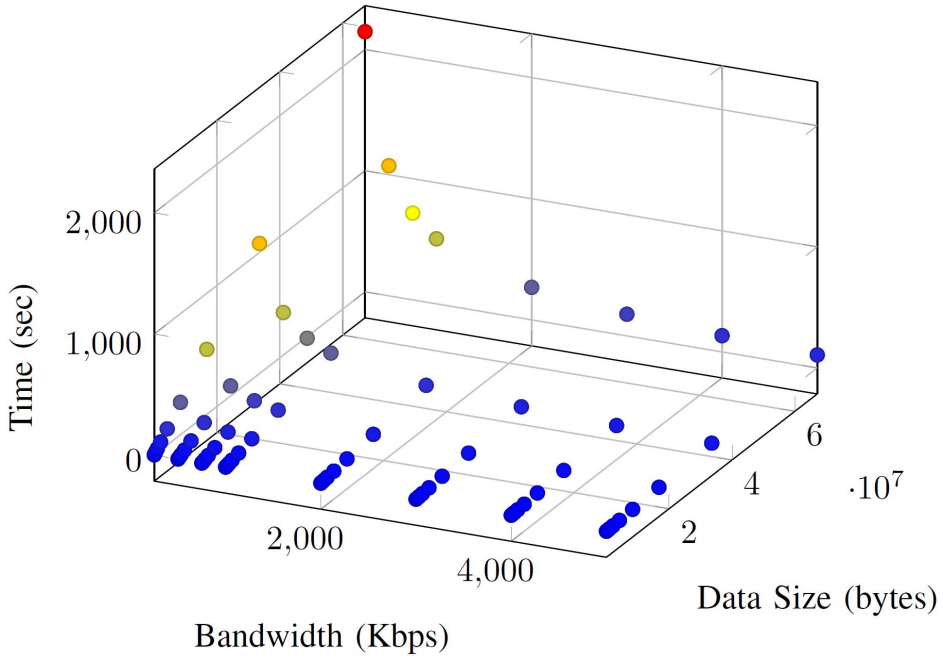
$$|x_1 - y_1| < |x_2 - y_2|.$$

Can we leverage KDN to construct accurate hyperprofiles for querying?

Collecting features to build our model

Visualization of network data from ns-3 simulations





Regression Model

Formally, we can represent the predicted variable as a linear function

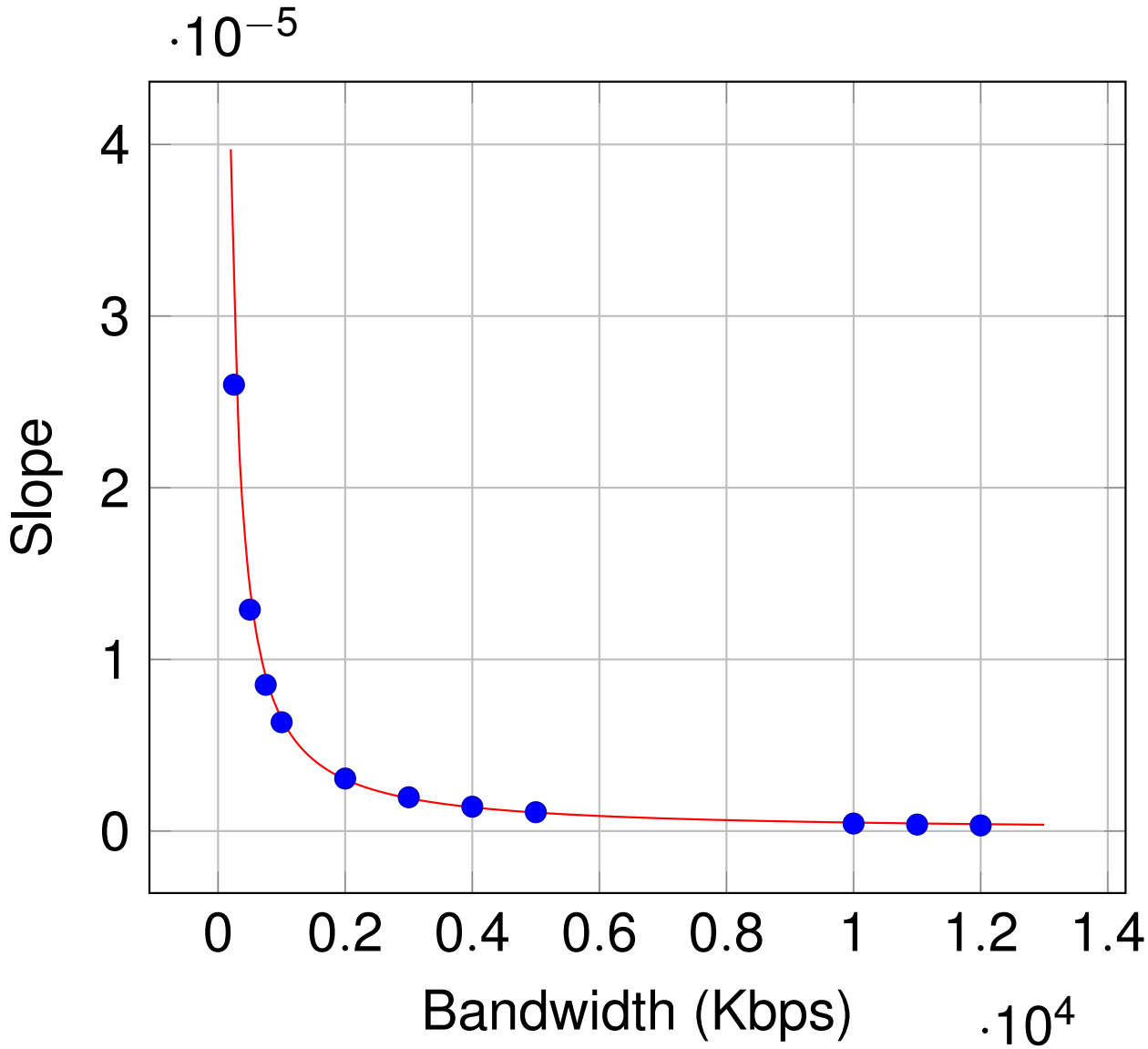
$$f_b(d_s) = m(b)d_s + c(b)$$

where the slope $m(b)$ and the y-intercept $c(b)$ are functions of bandwidth b.

In our study, f is either **energy** or **transmission time**.

Finding a line of best fit for our data

Graph of the function $m(b)$ on energy data

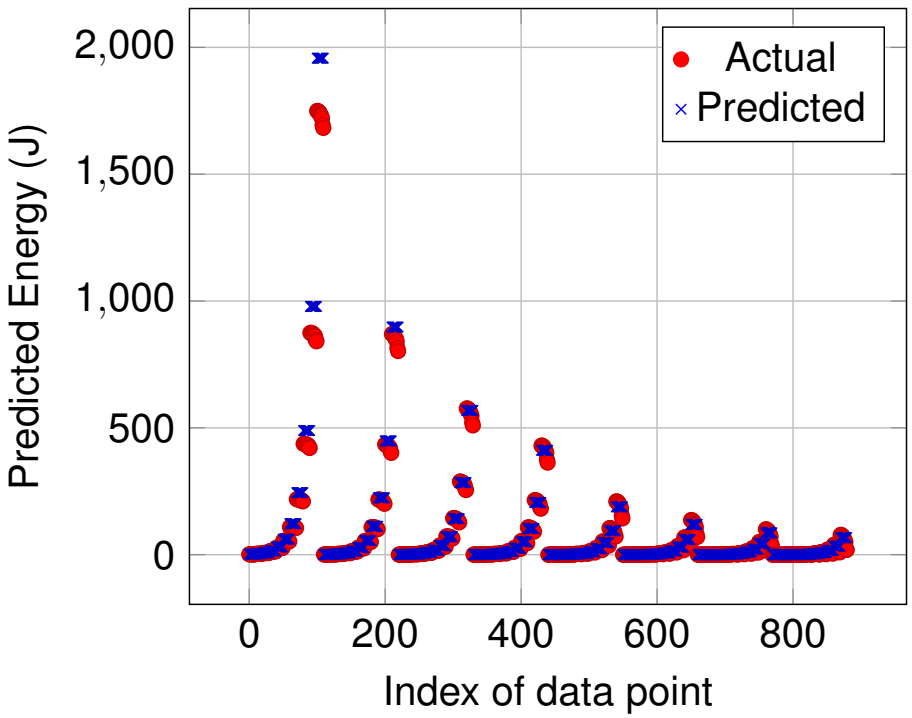


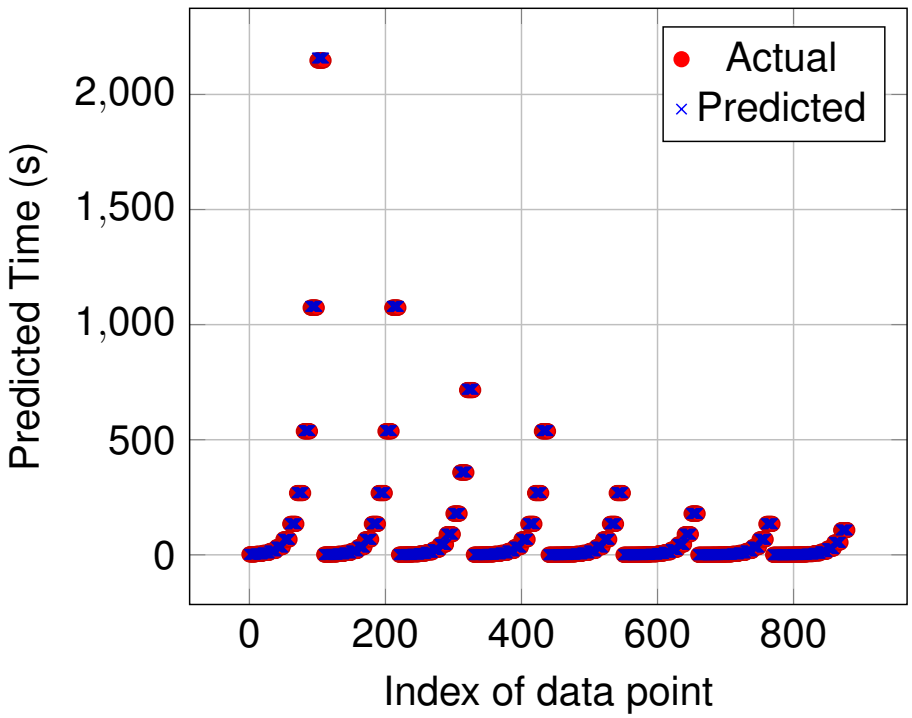
Regression Results

	Energy Consumption (e_c)	Time (t)
Bandwidth (b)	$m_1 = 0.015b^{-1.13}$ $R^2 = 0.997$	$m_2 = {}^{8.04 \cdot 10^6}/_b$ $R^2 = 1$ $c = 222873e^{0.0004b}$ $R^2 = 0.918$
Data Size (d_s)	$e_c = m_1 d_s$ Cross-validation: 0.99	$t = m_2 d_s + c$ Cross-validation: 0.99

Testing our model – varying physical distance

Comparing our model's predicted values versus actual values





Contributions

- We showed that network metrics can be encoded meaningfully into a multidimensional space
- Using machine learning to compute hyperprofiles in the knowledge plane is a viable approach to select nodes for computation offloading
- We investigated relevant data structures for *k*NN queries along with how *k*NN differs from other approaches

Future Work

- Setup an experiment to evaluate a hyperprofile based offloading scheme compared to standard schemes
  - offload images from a Google Glass device onto a set of servers running Tensorflow for object recognition
- Expand our idea of hyperprofile-based resource allocation to areas (e.g. routing) other than just computation offloading
- Explore the construction of feature spaces; investigate fitness profiles according to application requirements in edge networks

Thank You

Questions?

Our article:
Hyperprofile-based Computation Offloading for Mobile Edge Networks. arXiv preprint 1707.09422.