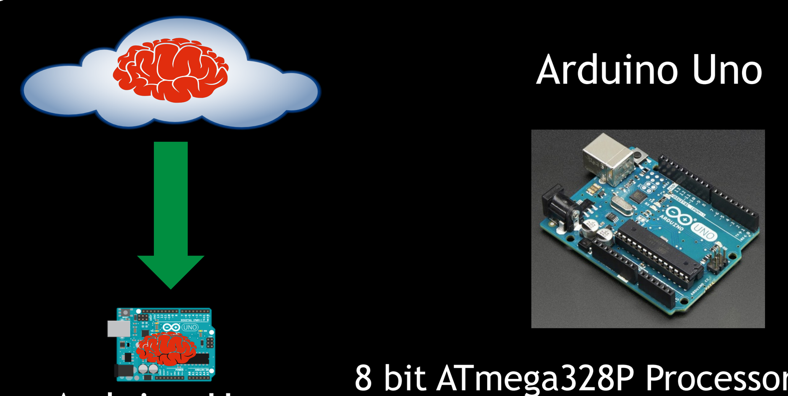
Efficient ML in 2 KB RAM for the Internet of Things

Ashish Kumar (MSR)
Saurabh Goyal (IIT Delhi)
Manik Varma (MSR)

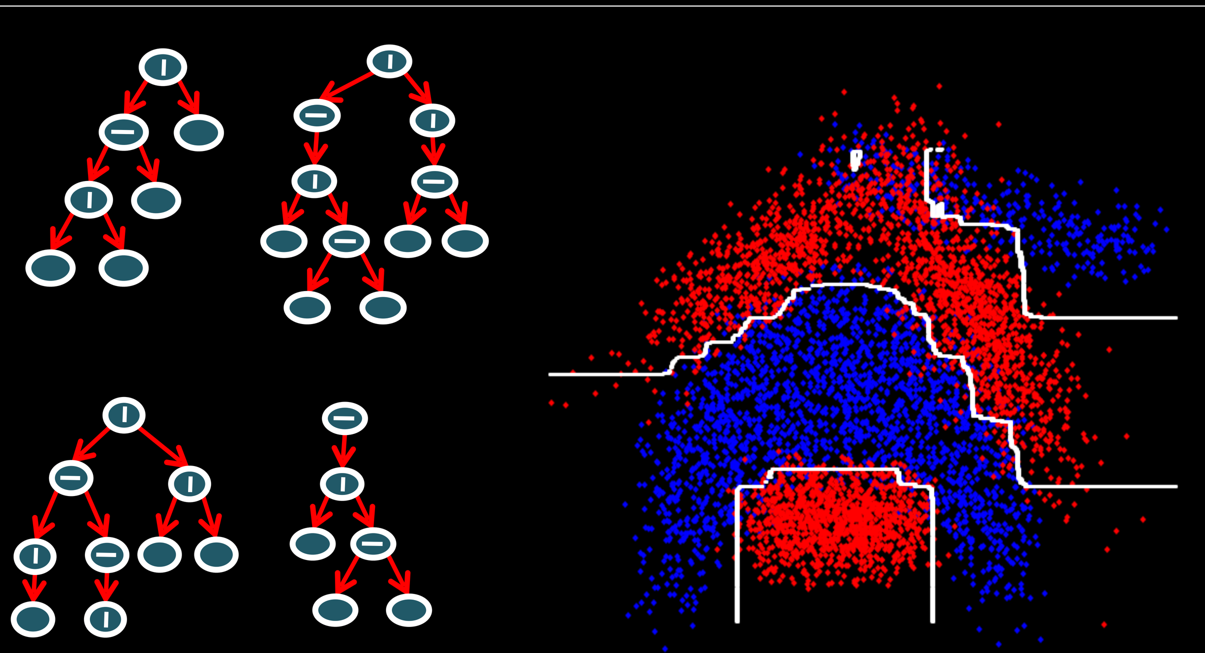
Objective

- To build an efficient tree classifier
  - Which can be trained on the cloud
  - But which can make predictions on tiny IoT devices



8 bit ATmega328P Processor at 16 MHz with 2 KB RAM & 32 KB read only Flash

Disadvantages of Tree Ensembles

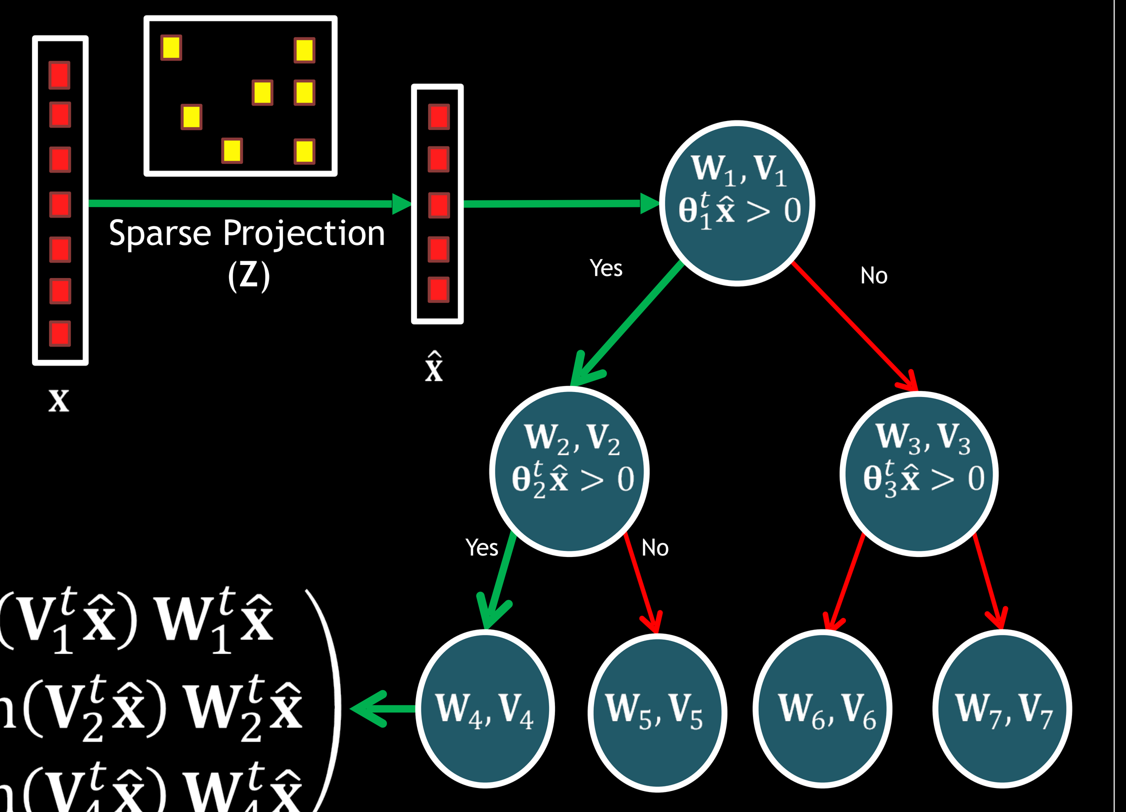


- Tree Ensembles might not fit in Kilobytes and might not be accurate

Bonsai - Key Ideas

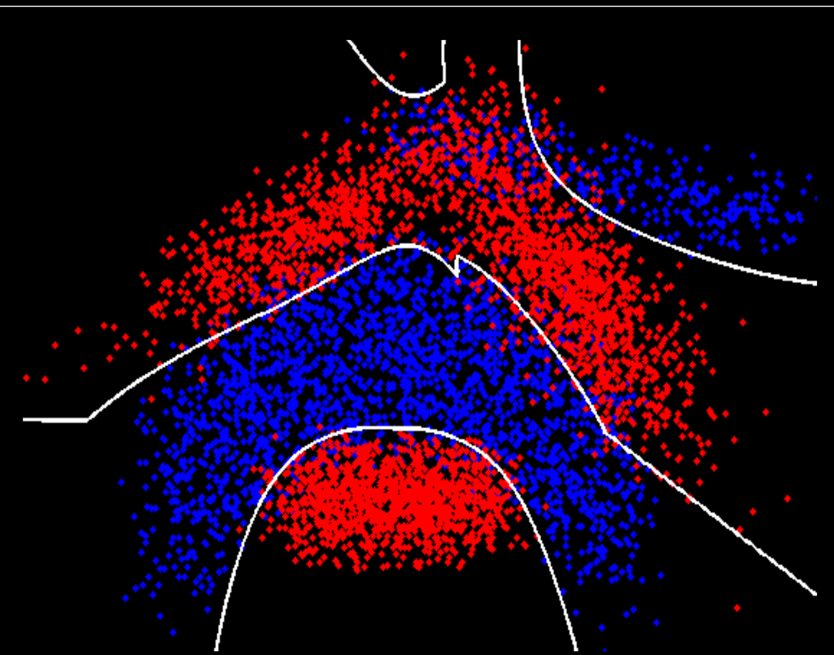
- We design Bonsai to be a single, shallow, sparse tree with powerful nodes for accurate prediction
- We reduce model size by learning Bonsai in a low-dimensional space into which all data is projected
- We jointly learn tree and projection parameters so as to maximize accuracy within the given budget

Bonsai - A Compact Tree Model



$$\hat{\mathbf{y}} = \begin{pmatrix} \tanh(\mathbf{V}_1^t\hat{\mathbf{x}})\,\mathbf{W}_1^t\hat{\mathbf{x}} \\ +\tanh(\mathbf{V}_2^t\hat{\mathbf{x}})\,\mathbf{W}_2^t\hat{\mathbf{x}} \\ +\tanh(\mathbf{V}_4^t\hat{\mathbf{x}})\,\mathbf{W}_4^t\hat{\mathbf{x}} \end{pmatrix}$$

Bonsai's Decision Boundaries



The Bonsai Objective Function

$$\mathrm{Min}_{\Theta,\mathbf{Z}}\ P = \frac{1}{2}Tr(\Lambda\Theta^\top\Theta) + \frac{\lambda}{2}Tr(ZZ^\top) + \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \hat{\mathbf{y}}_i; \Theta, \mathbf{Z})$$

$$\text{s.t.}\quad ||\Theta||_0 < B,\ ||\mathbf{Z}||_0 < S, \quad \Theta = [\mathbf{W}, \mathbf{V}, \theta]$$

- $\mathcal{L}$ is the loss function for classification, regression and ranking which can be optimized via SGD
- We place explicit budget constraints on the tree parameters Θ and sparse projection matrix $\mathbf{Z}$

Bonsai Optimization

Algorithm – Repeat the following steps till convergence

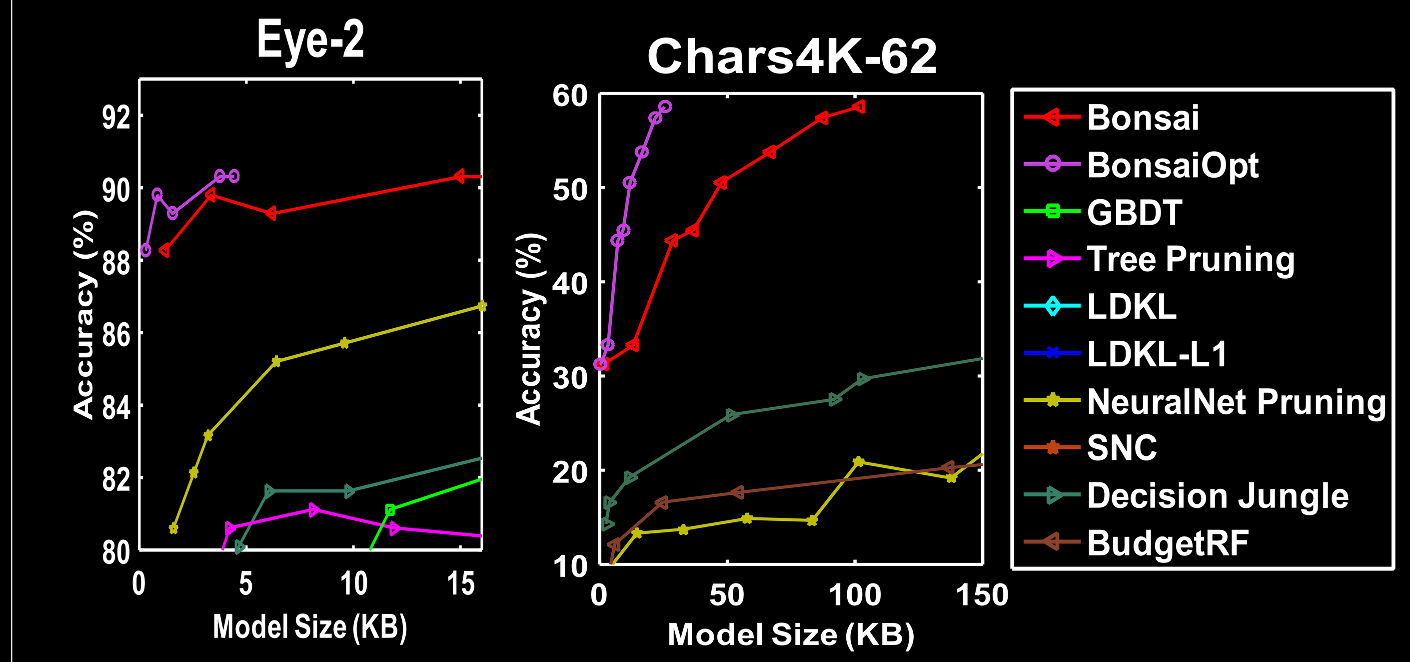
1. Mini batch gradient descent: K steps with fixed support

$$\Theta_{t+1} = \Theta_t - \eta_\Theta(\nabla_\Theta P)_{supp\{\Theta_t\}}$$
$$\mathbf{Z}_{t+1} = \mathbf{Z}_t - \eta_\mathbf{Z}(\nabla_\mathbf{Z} P)_{supp\{\mathbf{Z}_t\}}$$

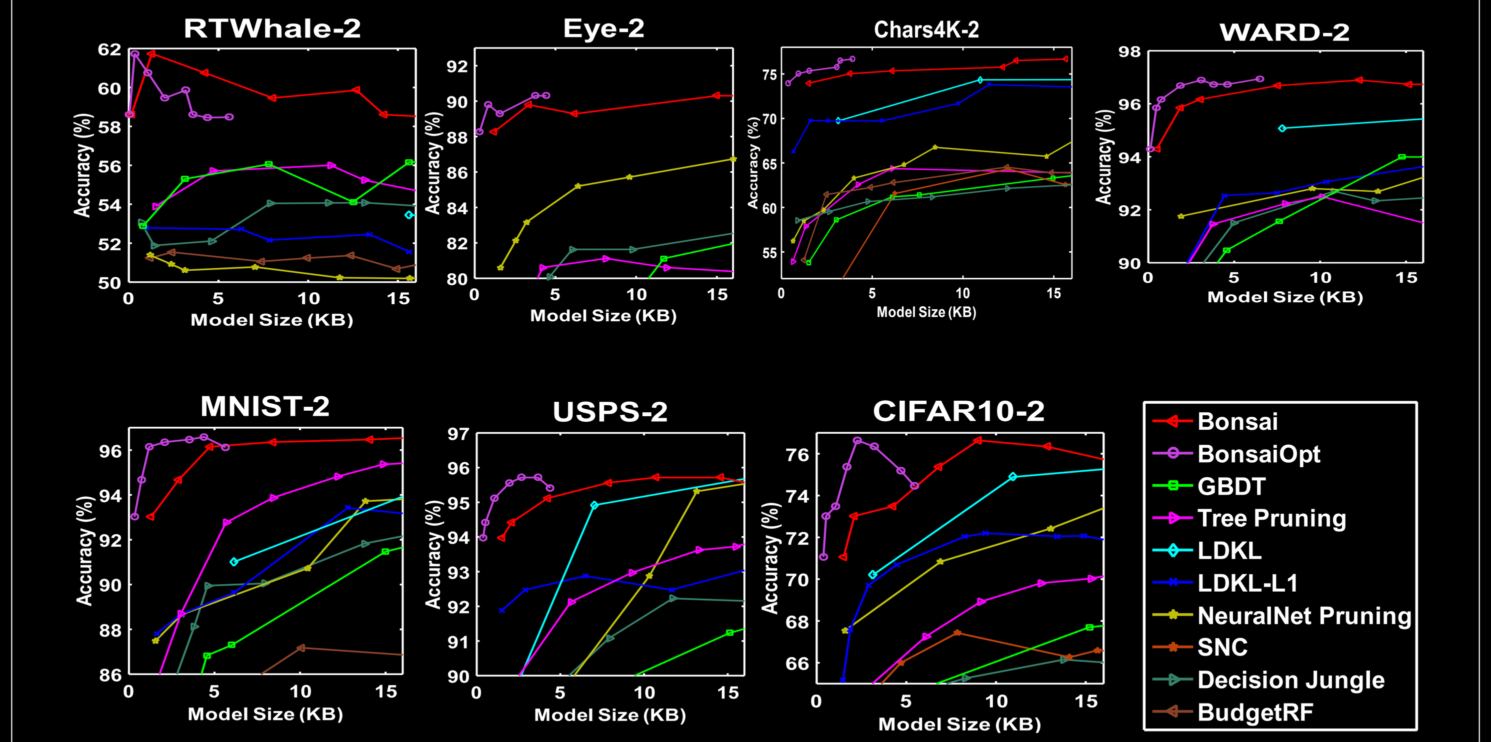
2. Hard thresholding step

$$\Theta_{t+1} = H_B(\Theta_t - \eta_\Theta \nabla_\Theta P)$$
$$\mathbf{Z}_{t+1} = H_K(\mathbf{Z}_t - \eta_\mathbf{Z} \nabla_\mathbf{Z} P)$$

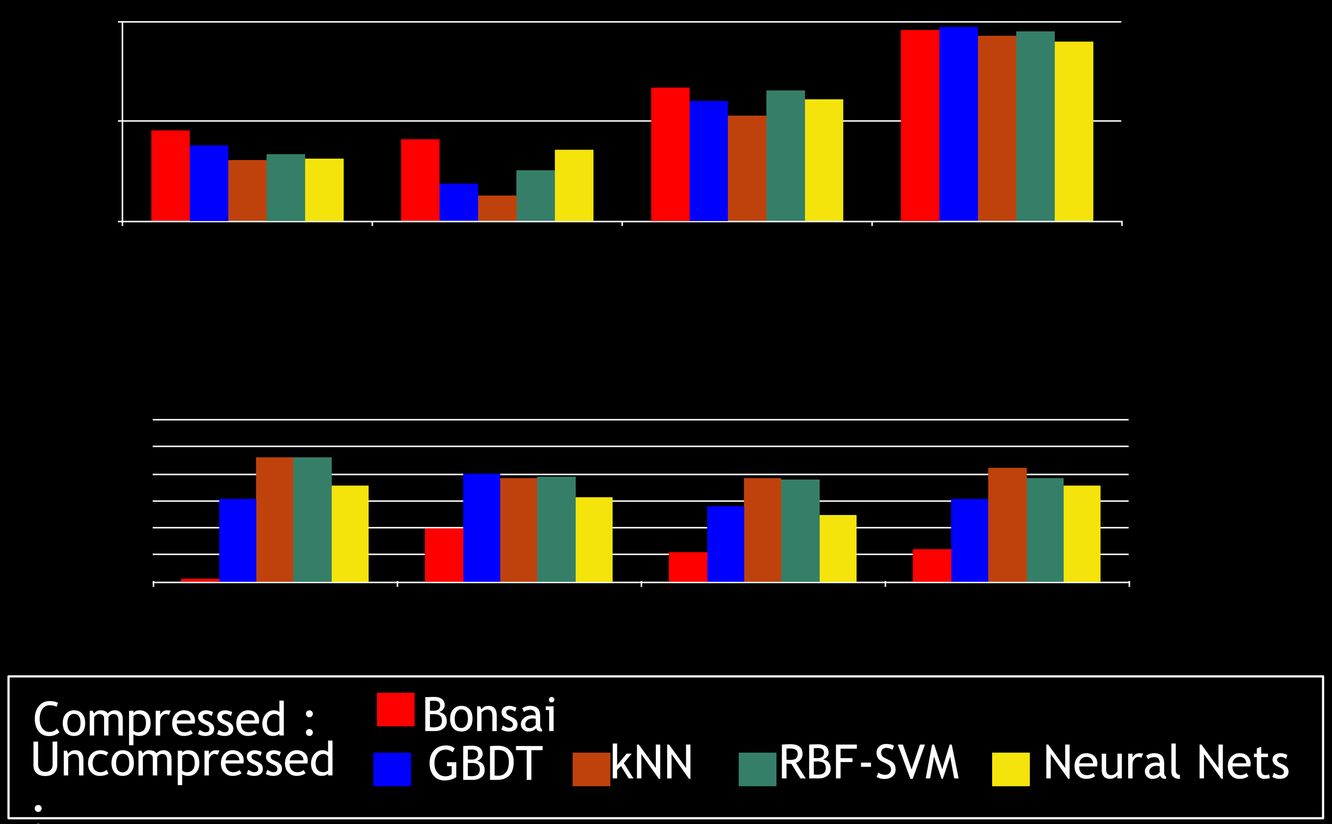
Prediction Accuracy vs Model Size



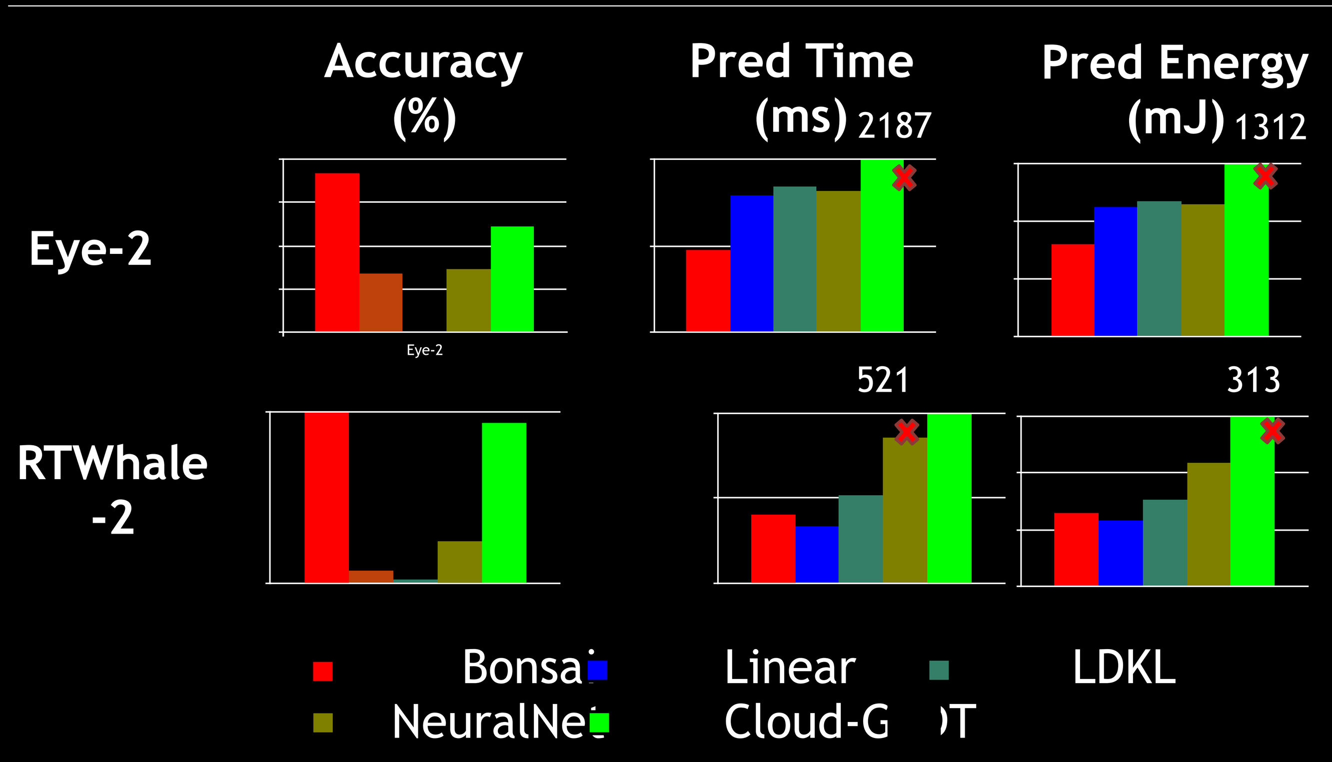
Prediction Accuracy vs Model Size



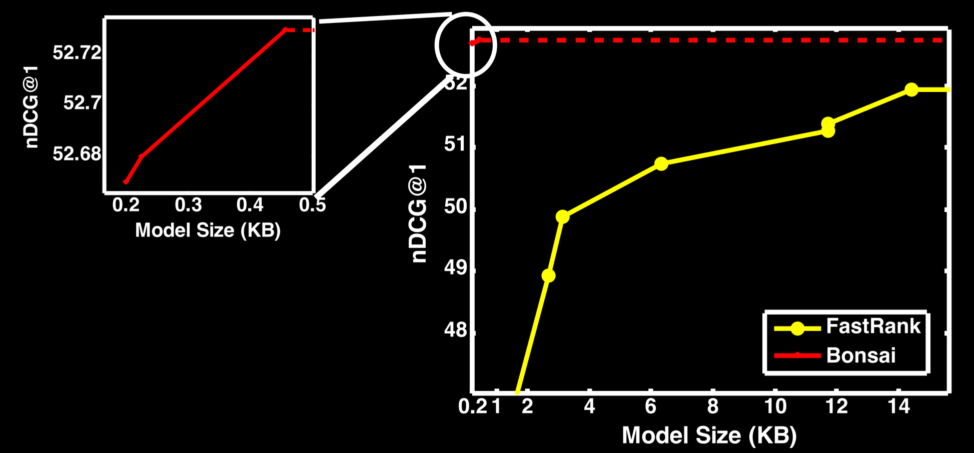
Comparison to Uncompressed Methods



Prediction Costs on the Arduino Uno



Bing L3/L4 Ranker Results



Code for Bonsai code can be downloaded from http://manikvarma.org/