IEEE WCCI 2020

IEEE World Congress on Computational Intelligence

Virtual Conference – July 19-24, 2020

Multi Objective Optimization of Multi Output Neural Trees

Varun Ojha, University of Reading

Giuseppe Nicosia, University of Cambridge

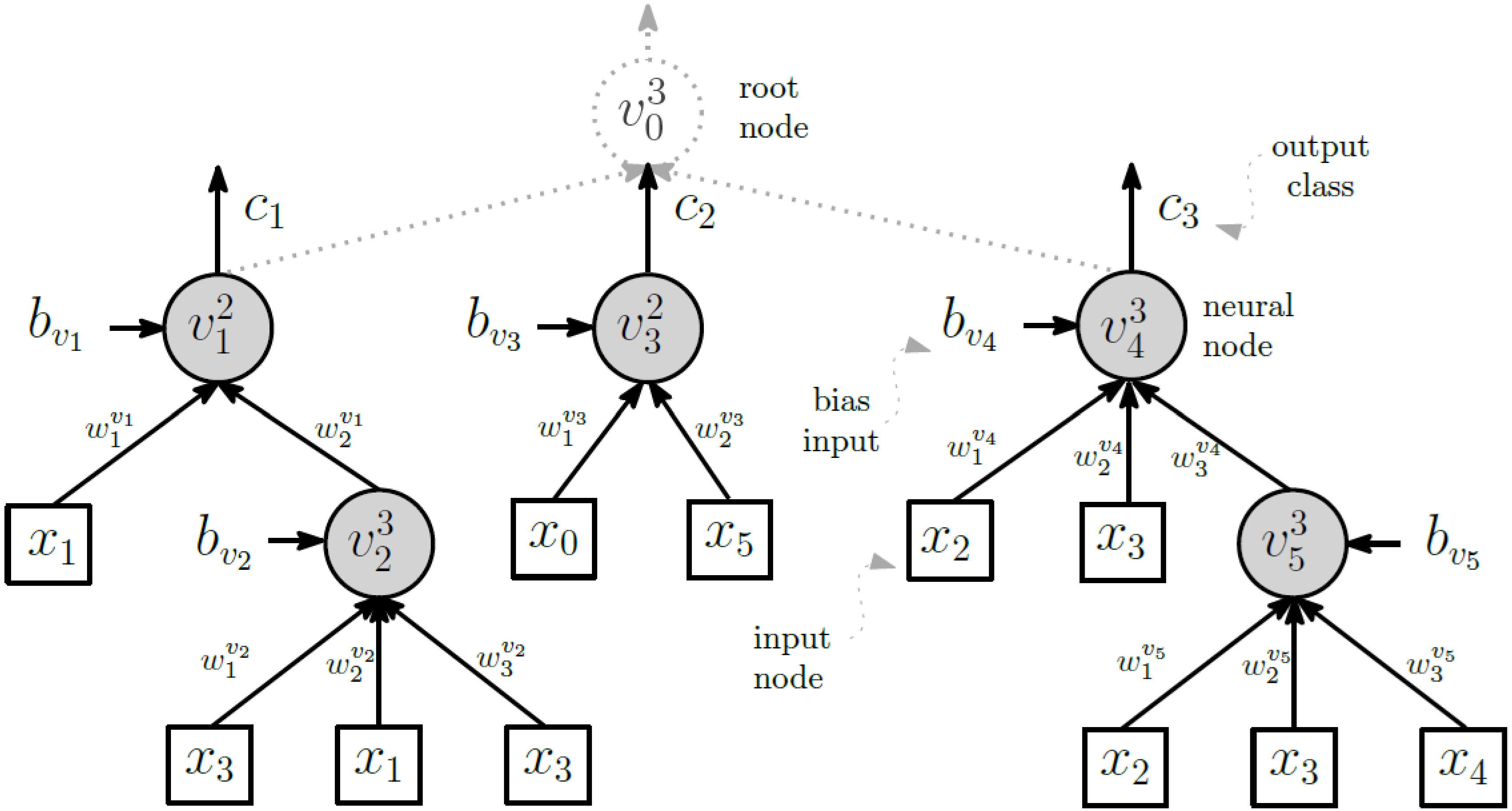
Contributions

- A new algorithm Neural Architectural Search called Multi-output Neural Tree (MONT) is designed for classification tasks

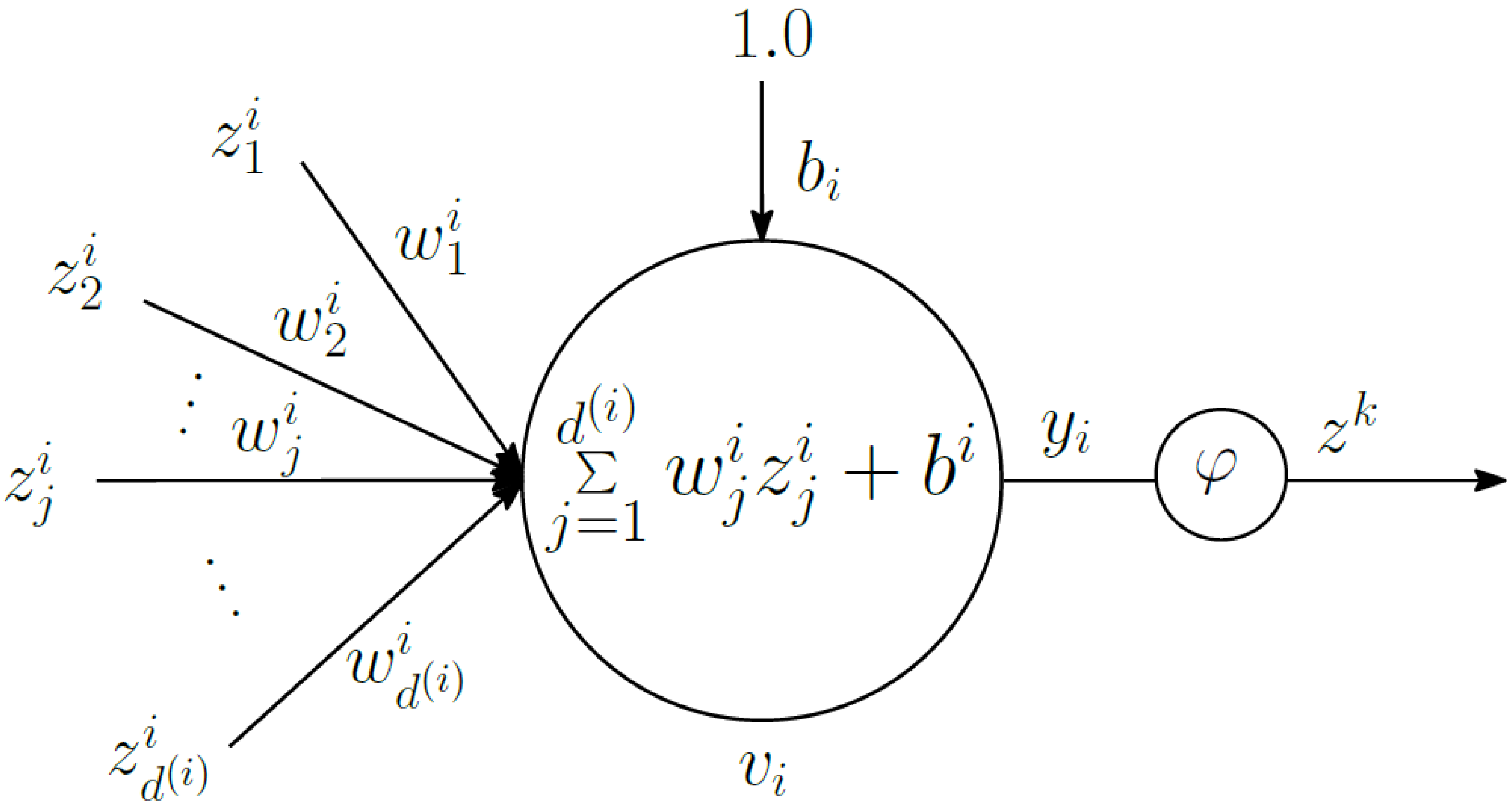
- A Pareto-optimality of evolutionary learning processes was investigated using hypervolume indicator analysis.

- A comprehensive analysis of the MONT's (trained with NSGA-III) performance compared with other algorithms and with MONT's other two training version GP, NSGAII is presented.

Multi Output Neural Tree



Neural Node



1.0
b_i
z_1^i
w_1^i
z_2^i
w_2^i
w_j^i
z_j^i
$\sum_{j=1}^{d(i)} w_j^i z_j^i + b^i$
y_i
φ
z^k
$w_{d(i)}^i$
$z_{d(i)}^i$
v_i

Multi Output Tree Properties

- Minimum Node $n \leq \frac{(m^{p+1}-1)}{m-1}$, where $m \geq 2$ is max child per node and p is the depth

- Time Complexity is O(n), where n is the number of nodes in the tree.

- Max possible combination of tree Architecture is roughly close to Catalan number $C_n = \frac{1}{[(m-1)(n+1)]} . \binom{m.n}{n}$

Multiple Objectives

- Minimization of miss classification rate:

$$\boldsymbol{f_1 = \frac{1}{N}\sum_{i=1}^{N}(\hat{y} \neq y)}$$

where $\hat{y}$ is predicted out of the tree and y is the target output, N is the number of examples.

- Minimization of Tree size

$$\boldsymbol{f_2 = Tree\ Size}$$

Multi Objective Optimization

Algorithm 1 Evolutionary Learning of MONT

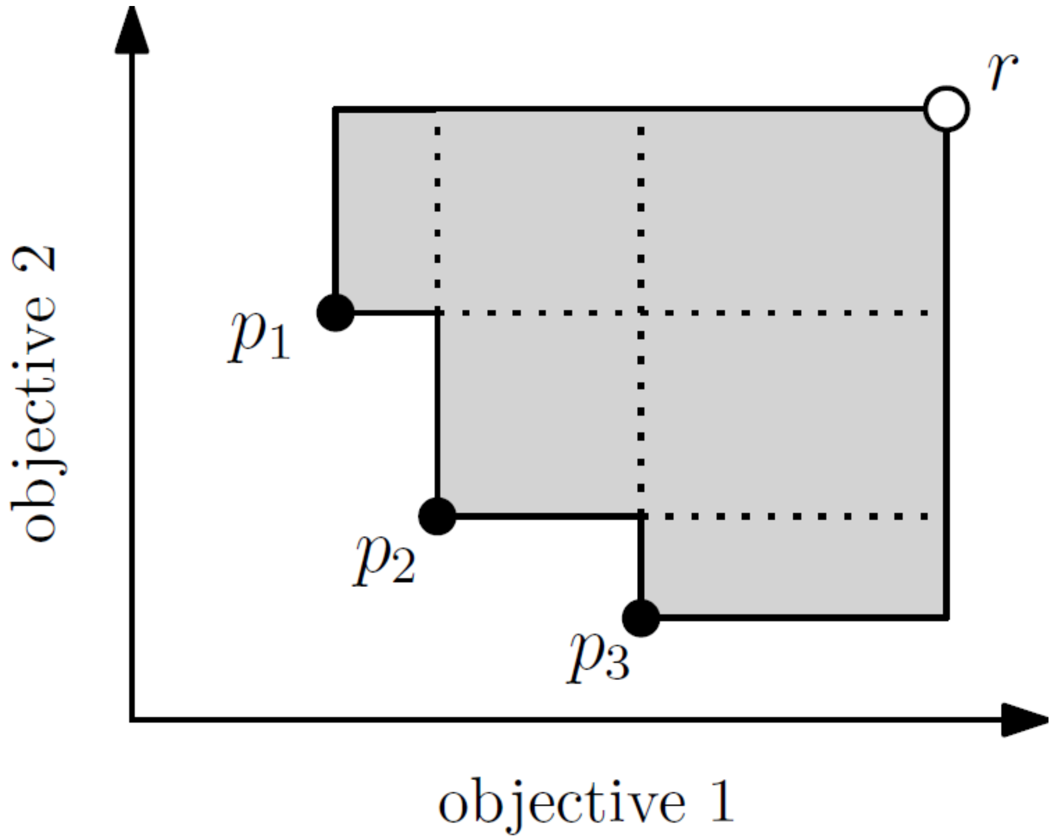
Require: Initial population P_0 of randomly generated neural trees, objectives $\mathcal{F} = [f_1, f_2]$, data $\mathcal{S}$, maximum evolutionary generations (termination criteria) g_{max}.

Ensure: Final population $P_{g_{max}}$ of Pareto-optimal trees

```
function TREE EVOLUTION(P_0, F, S, g_max.)
    while number of generation g reached g_max do
        selection: parent trees for crossover and mutation
        generation: a new population Q
        combined population: R = P_g + Q
        evaluation: NSGA-II/III non-dominated sorting(R)
        survive: elitism/niching (P_{g+1}, size(P_0), R)
    end while
    return P_{g_max}
end function
```

Hypervolume Indicator Analysis for Pareto-Optimality

- Algorithm that cover larger area is better.
- Pareto points are p_i for objectives 1 and 2
- Reference point is indicated by r



Quality of Hypervolume Indicator H_i On Three Versions of MONT

- MONT_1 = Genetic Algorithm
- MONT_2 = NSGA – II
- MONT_3 = NSGA – III

QUALITY OF TRADE-OFF OBTAINED BY HYPERVOLUME INDICATOR H_i ON THREE VERSIONS OF MONT.

Data	MONT_1	MONT_2	MONT_3
aus	84.10	83.57	83.57
gls	44.14	56.16	52.35
hrt	78.74	77.46	77.46
ion	85.65	83.83	88.97
irs	85.60	90.00	89.97
pma	73.38	76.01	76.01
vhl	49.21	52.72	51.99
win	86.47	87.44	87.28
wis	90.68	90.55	91.34
Avg.	**75.33**	**77.53**	**77.66**

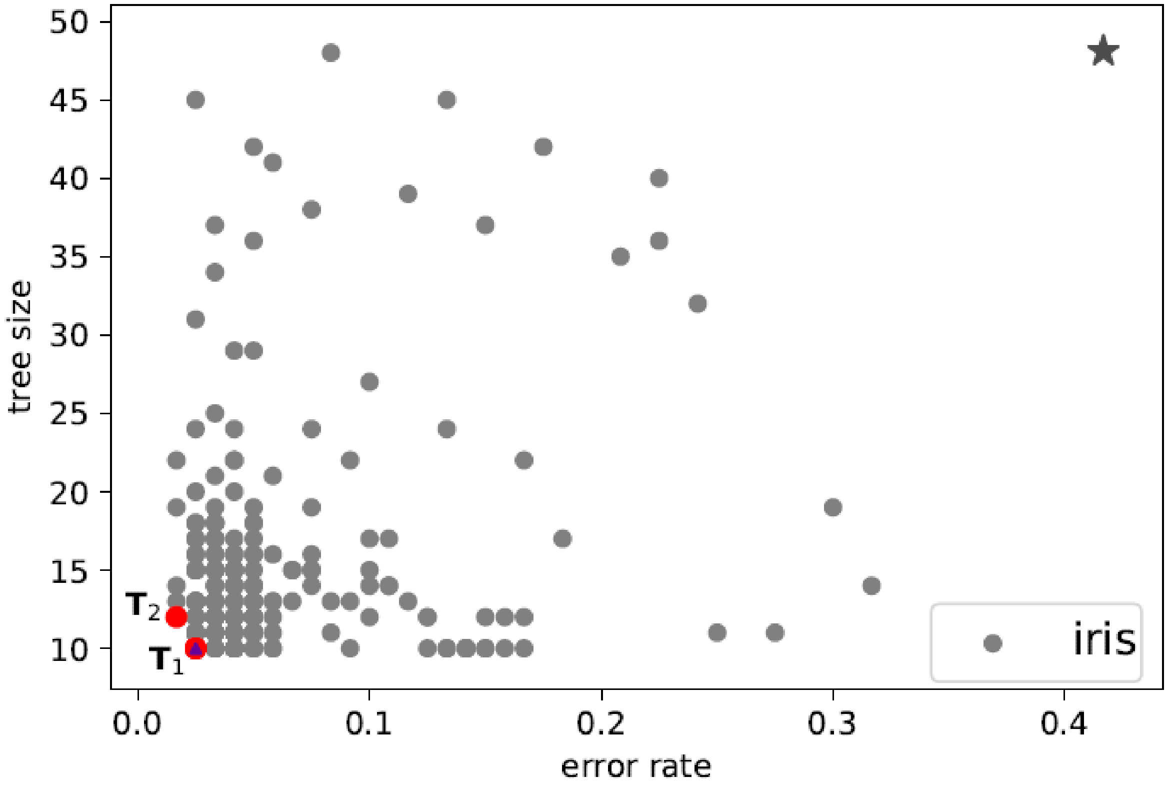
Classification Results

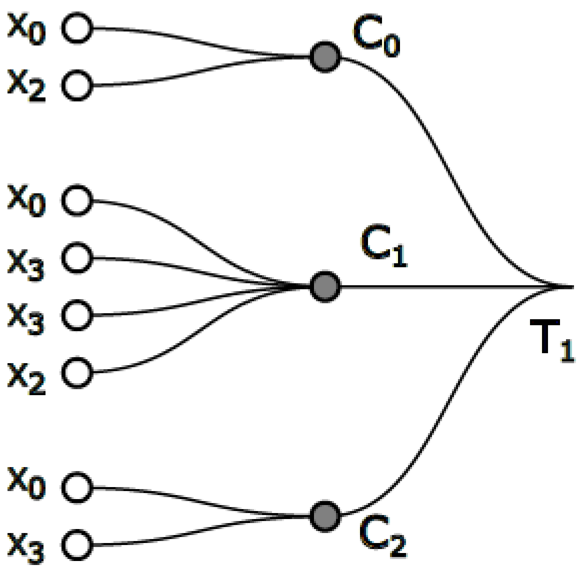
AVERAGE TEST ERROR-RATE F_μ AND VARIANCE F_σ OF 30 RUNS OF EXPERIMENTS ON MONT$_3$ AND OTHER ALGORITHMS

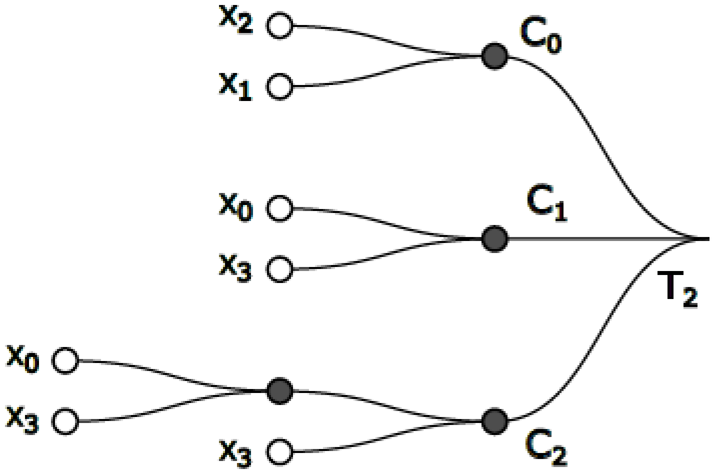
Algorithm	f_1	aus	hrt	ion	pma	wis	irs	win	vhl	gls	Avg.
		data									
MONT$_3$	f_μ	**0.111**	**0.191**	**0.102**	**0.201**	**0.038**	**0.011**	**0.048**	**0.450**	0.371	**0.169**
	f_σ	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.001	0.021
HFNT	f_μ	0.174	0.230	0.178	0.284	0.065	0.189	0.176	0.591	0.601	0.276
	f_σ	0.006	0.004	0.003	0.003	0.001	0.019	0.014	0.005	0.015	0.039
MLP	f_μ	0.175	**0.213**	**0.094**	**0.249**	**0.024**	**0.040**	**0.037**	**0.183**	0.367	**0.154**
	f_σ	0.001	0.004	0.001	0.001	0.001	0.002	0.000	0.001	0.004	0.013
REP-T	f_μ	**0.150**	0.247	0.107	0.255	0.096	0.064	0.071	**0.291**	**0.348**	**0.181**
	f_σ	0.001	0.004	0.002	0.001	0.003	0.001	0.000	0.001	0.005	0.012
NBC	f_μ	0.231	**0.176**	0.166	**0.244**	**0.026**	0.047	**0.070**	0.544	0.525	0.225
	f_σ	0.001	0.003	0.002	0.001	0.001	0.001	0.001	0.001	0.008	0.035
DT	f_μ	**0.146**	0.312	0.126	0.337	0.514	0.070	0.370	0.463	**0.337**	0.297
	f_σ	0.001	0.012	0.001	0.001	0.004	0.001	0.002	0.002	0.006	0.024
SVM	f_μ	0.455	0.461	**0.073**	0.353	0.532	**0.029**	0.369	0.754	**0.353**	0.376
	f_σ	0.002	0.004	0.001	0.001	0.006	0.001	0.002	0.000	0.004	0.046

Note: for all datasets three lowest average test error rates are marked in Bold.

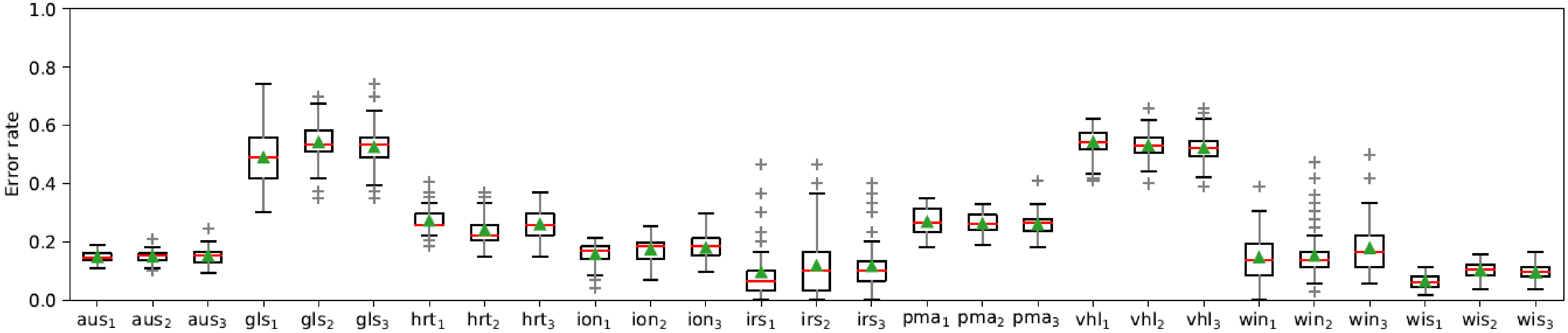
Example Tree Selection







Tree Best Activation Function



Gaussian, **sigmoid (evaluated best)**, and tanh respectively marked 1, 2, and 3 as the subscript of dataset names.

Question?

v.k.ojha@reading.ac.uk