Contractor Disclosure Form 712A – Deadline: 2 June 2014

MORS Symposium

16-19 June 2014, Hilton Mark Center, Alexandria, VA

Fax completed form to 703-933-9066 or email to liz@mors.org

Abstract 601

PART I **Author Request** - The following author(s) request authority to disclose the following presentation at the MORS Symposium with subsequent publication in the MORS Final Report, and posting on the MORS website, if applicable.

Principal Author: Thomas A Donnelly

Other Author(s): N/A

Principal Author's **Organization** and complete mailing address:
SAS Institute Inc.
27 Farmingdale Ln
Newark, DE 19711

Principal Author's Signature: X Thomas A. Donnelly Date: 28 May 2014

Phone: 302-489-9291 FAX: 919-677-4444

Email: tom.donnelly@jmp.com

Title of Presentation:
Improving Prediction of Cyber Attacks Using Ensemble Modeling

This presentation is: ☐ SECRET ☐ SECRET//REL TO FVEY ☐ CONFIDENTIAL ☐ CONFIDENTIAL//REL TO FVEY ■ UNCLASSIFIED ☐ Other ________ and will be presented in:

☐ Tutorial ■ List all WG(s) #: WG-5

This work was performed in connection with a government contract.	☐ YES (Complete Parts I, II, & III)
This presentation is based on material developed by the author as part of company-approved research e.g. IR&D and was NOT done under a government contract.	☐ YES (Complete Parts I, II & III)
This presentation was NOT done under a government contract, contains no government information, is my own work and is approved for public release.	■ YES (Complete Part I only)

PART II **Contractor Security Officer Endorsement** - The Contractor Security Officer concurs in the assigned classification and consents to the disclosure. A copy of the presentation, as made, will be provided to the Contracting Officer for approval.

IMPROVING PREDICTION OF CYBER ATTACKS USING ENSEMBLE MODELING

June 17, 2014
82nd MORSS
Alexandria, VA

Tom Donnelly, PhD
Systems Engineer & Co-insurrectionist
JMP Federal Government Team

ABSTRACT

Improving Prediction of Cyber Attacks Using Ensemble Modeling

In 1998 DARPA developed a representative cyber-attack data set with over 20 attack types, 41 potentially causal factors, and nearly 5 million rows of data. These and derivative data are analyzed using a variety of predictive models, including nominal logistic, decision trees, and neural models. It will be shown that the ability to predict attacks can be further improved by averaging models. Both simple algebraic averaging of model probabilities as well "ensemble modeling" - where models are used as inputs to other models - will be demonstrated.

OUTLINE

- Goals
- Background
- Approaches and Strategies
- Model Averaging
- Visualize Results
- Summary

GOALS

- Take “Data Mining Challenge” data set and develop best predictor model
- Learn about different approaches to data mining and model averaging

ORIGINAL KDD DATA SET

TABLE I

STATISTICS OF REDUNDANT RECORDS IN THE KDD TRAIN SET

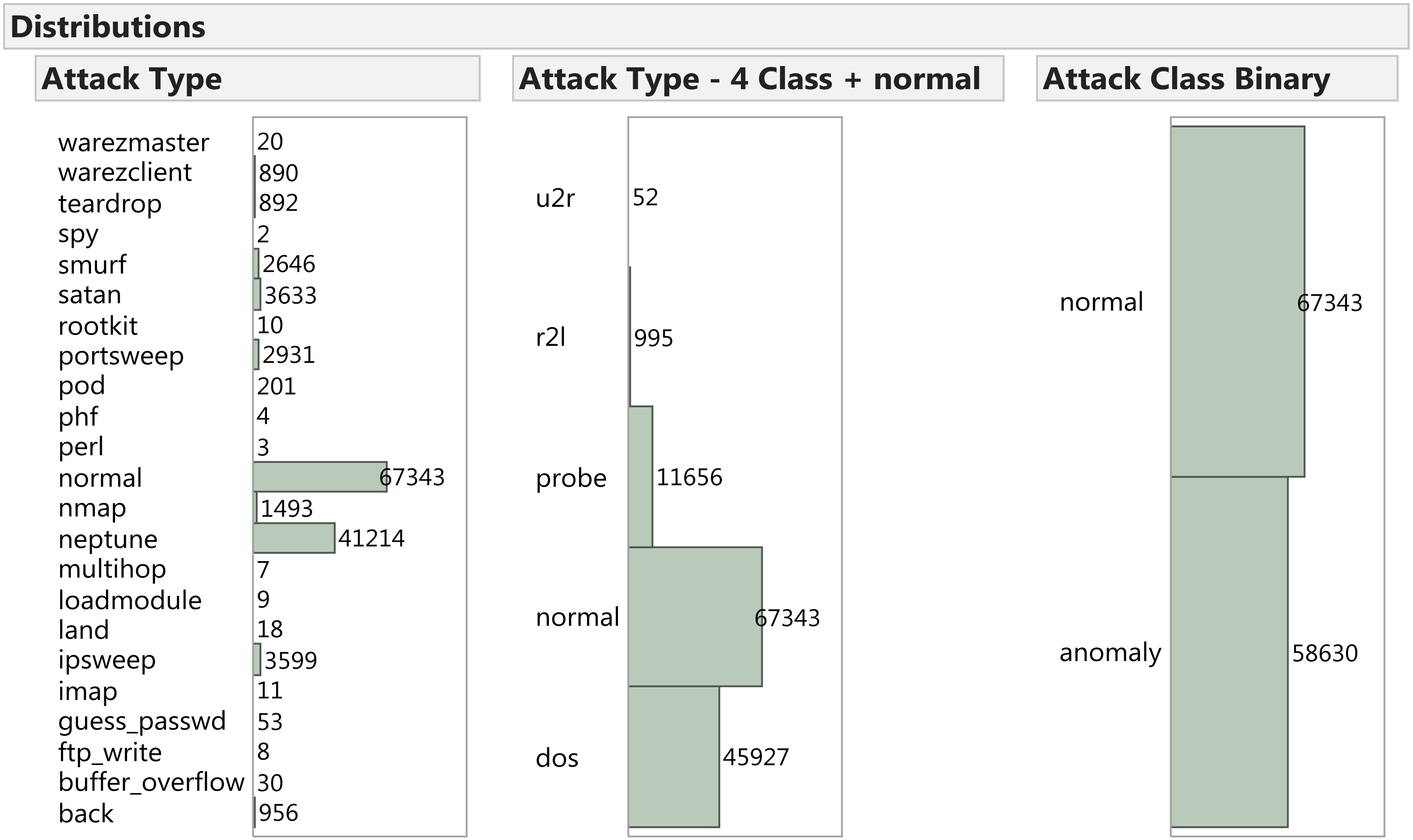
	Original Records	Distinct Records	Reduction Rate
Attacks	3,925,650	262,178	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%

TABLE III

STATISTICS OF RANDOMLY SELECTED RECORDS FROM KDD TRAIN SET

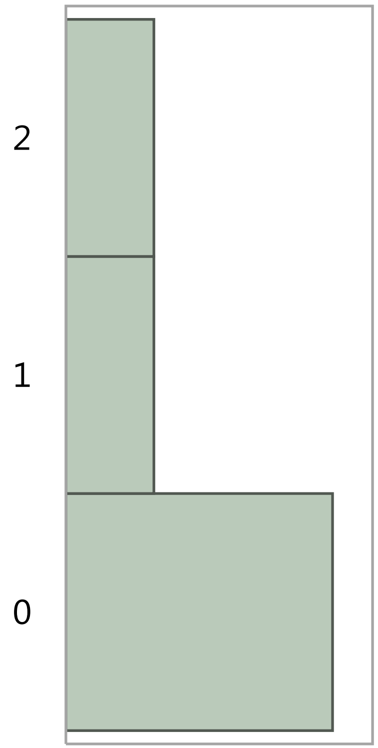
	Distinct Records	Percentage	Selected Records
0-5	407	0.04	407
6-10	768	0.07	767
11-15	6,525	0.61	6,485
16-20	58,995	5.49	55,757
21	1,008,297	93.80	62,557
Total	1,074,992	100.00	125,973

ATTACK TYPE BINNING



RANDOM HOLDBACK SUBSETS 60% TRAIN = 0, 20% VALIDATE = 1, AND 20% TEST = 2

Holdback



Frequencies

Level	Count	Prob
0	75584	0.60000
1	25194	0.20000
2	25195	0.20000
Total	125973	1.00000

N Missing 0

3 Levels

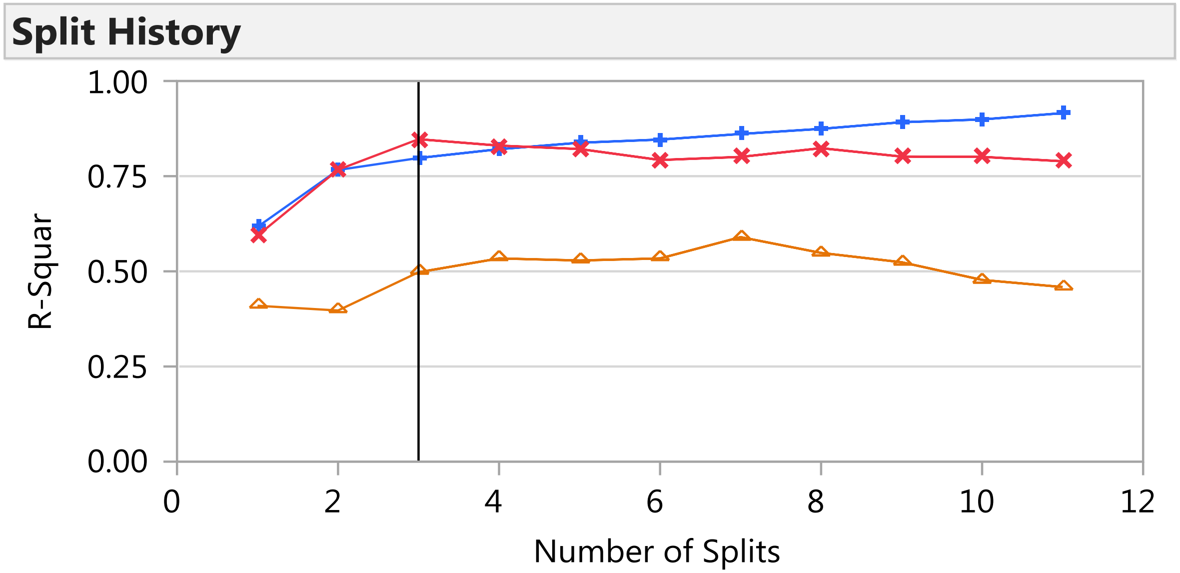
The Elements of Statistical Learning – Data Mining, Inference, and Prediction

Hastie, Tibshirani, and Friedman – 2001

(Chapter 7: Model Assessment and Selection)

HONEST ASSESSMENT APPROACH USING TRAIN, VALIDATE (TUNE), AND TEST SUBSETS

R-SQUARE VS. #SPLIT DECISION TREE MODE

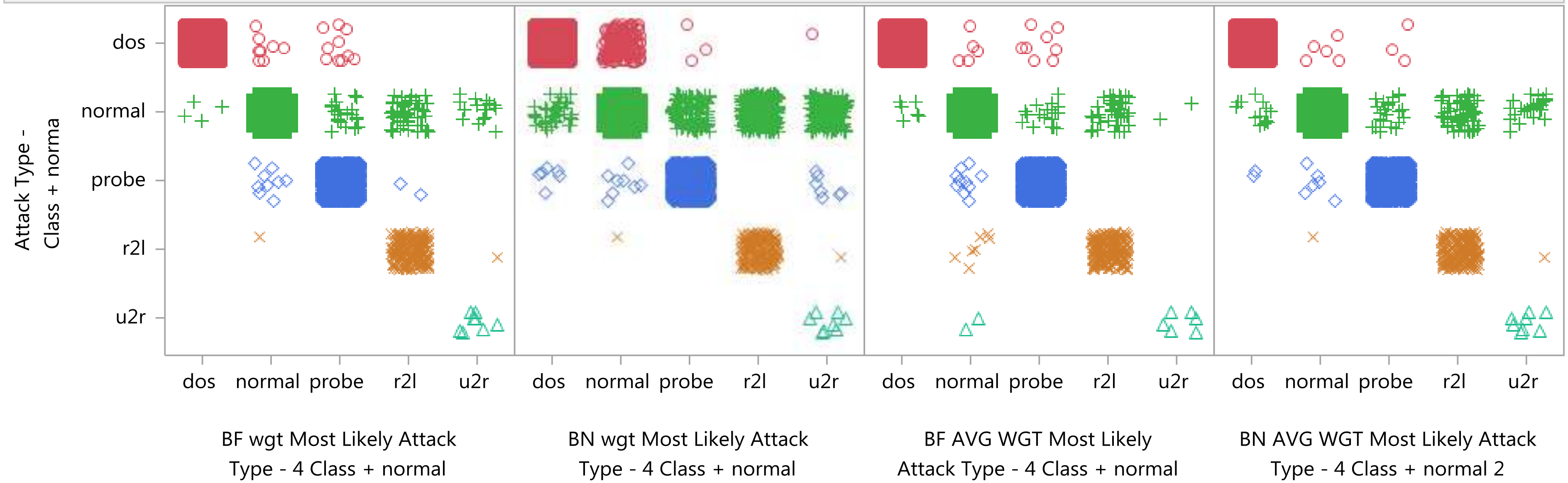


Validation Data in Red

Test Data in Orange

ACTUAL VS. PREDICTED FOR TEST SUBSET FOR FOUR MODELS USING ALL 41 FACTORS

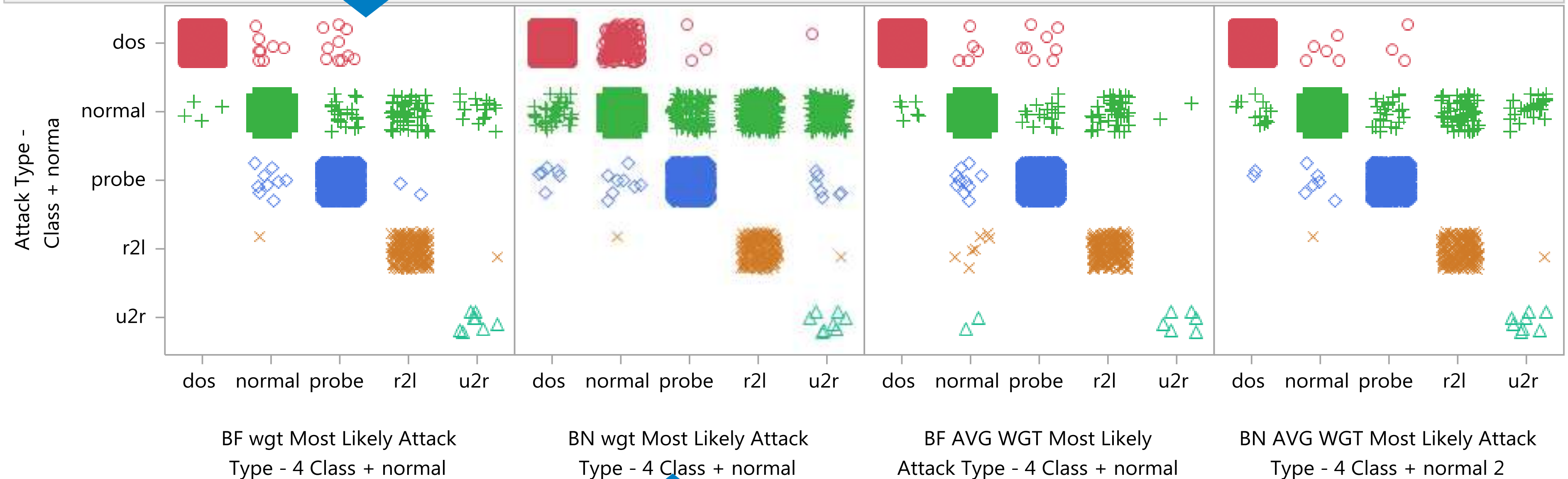
Scatterplot Matrix Holdback=2



DECISION TREE (BF) AND NEURAL NET (BN)

	Holdback				
	2				
	BF wgt Most Likely Attack Type - 4 Class + norma				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9212	8	10	0	0
normal	4	13289	21	53	17
probe	0	9	2381	2	0
r2l	0	1	0	179	1
u2r	0	0	0	0	8

Scatterplot Matrix Holdback

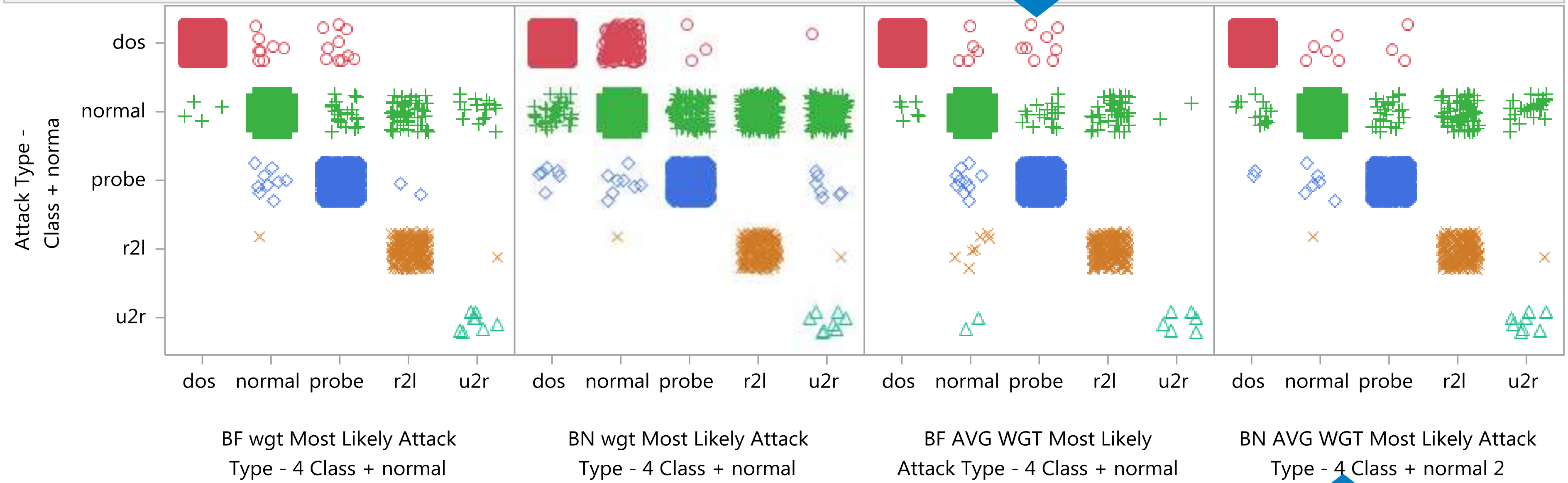


	Holdback				
	2				
	BN wgt Most Likely Attack Type - 4 Class + norma				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9074	152	3	0	1
normal	41	12735	126	329	153
probe	6	8	2371	0	7
r2l	0	1	0	179	1
u2r	0	0	0	0	8

ENSEMBLE MODELS

	Holdback				
	2				
	BF AVG WGT Most Likely Attack Type - 4 Class + norma				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9217	5	8	0	0
normal	6	13317	18	41	2
probe	0	10	2382	0	0
r2l	0	7	0	174	0
u2r	0	2	0	0	6

Scatterplot Matrix Holdback=2



OUTPUTS OF FIRST TWO MODELS USED AS INPUTS FOR LAST TWO MODELS

	Holdback				
	2				
	BN AVG WGT Most Likely Attack Type - 4 Class + normal				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9222	5	3	0	0
normal	13	13255	23	66	27
probe	2	6	2384	0	0
r2l	0	1	0	179	1
u2r	0	0	0	0	8

APPROACHES & STRATEGIES

- “Honest Assessment” Approach – Divide Data into Train, Validate, & Test Sets
- Model 4 Largest of 22 Attack Types plus Normal
- Weight attack types by the inverse of their probability of occurrence so that rare events get more weight than common attacks
- Initial Analyses - Model with ALL 41 factors
- Use many types of models and select better ones to average
  - Partition and Bootstrap Forest decision trees - (BF was better)
  - Single-Layer, Dual-Layer, and Boosted (sequential) Neural Nets – (BN was best)
- Later Analyses - Down select to more critical few factors - 11 chosen using Bootstrap Forest decision tree method
- Add 3 factors consisting of random data (Normal, Uniform, Integer)
- Stratify attack Types by Train-Validate-Test subsets
- Model the Bias – increase weight of misclassified cases (“Nate Silver” approach)

DECISION TREES

- Also known as Recursive Partitioning, CHAID, CART
- Models are a series of nested IF() statements, where each condition in the IF() statement can be viewed as a separate branch in a tree.
- Branches are chosen so that the difference in the average response (or average response rate) between paired branches is maximized.
  - For all factors bin factor values or levels into two buckets such that the means of the two buckets are as far apart as possible.
  - Split on factor with the biggest difference in bucket means.
- Tree models are "grown" by adding more branches to the tree so the more of the variability in the response is explained by the model

DECISION TREE STEP-BY-STEP

RSquare
0.000

All Rows

Count	G^2
90	77.800668

Level	Rate	Prob
Accep	0.8444	0.8444
Reject	0.1556	0.1556

Goal is to predict "Rejects" & "Accepts""

Overall Accept Rate is 84.44%
Overall Reject Rate is 15.56%

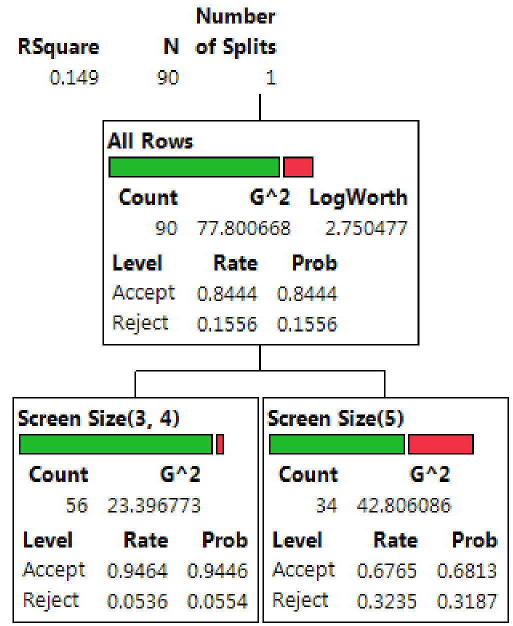
Candidates

Term	Candidate G^2		LogWorth	Cut Point
API Particle Size	4.04050319		0.986886932	Small,Large
Mill Time	10.63219688		1.912625603	11
Screen Size	11.59780917	>	2.750476973	3,4
MgSt Supplier	1.99715970		0.802459554	Jones Inc
Lactose Supplier	1.07597470		0.523458492	James Ind
Sugar Supplier	3.99502860		1.340705011	Sour
Talc Supplier	0.00000000		0.000000000	Rough
Blend Time	2.46622023		0.066048548	15.887
Blend Speed	6.86574102		0.717212865	60.772
Compressor	0.00153207		0.013776004	COMPRESS
Force	7.53188562		0.855446810	24.691
Coating Supplie	0.82675321		0.217072294	Mac
Coating Viscosit	4.66879353		0.322714711	96.413
Inlet Temp	7.28399996		0.803171227	106.39
Exhaust Temp	7.17119361		0.779703315	68.592
Spray Rate	15.01998363	<	2.736639439	403.26
Atom. Pressure	3.36570749		0.149475063	58.787

Candidate "X's"

- Search through each of these
- Examine Splits for each unique level in each X
- Find Split that maximizes "LogWorth"
    - Will find split that maximizes difference in proportions of the target variable

DECISION TREE STEP-BY-STEP



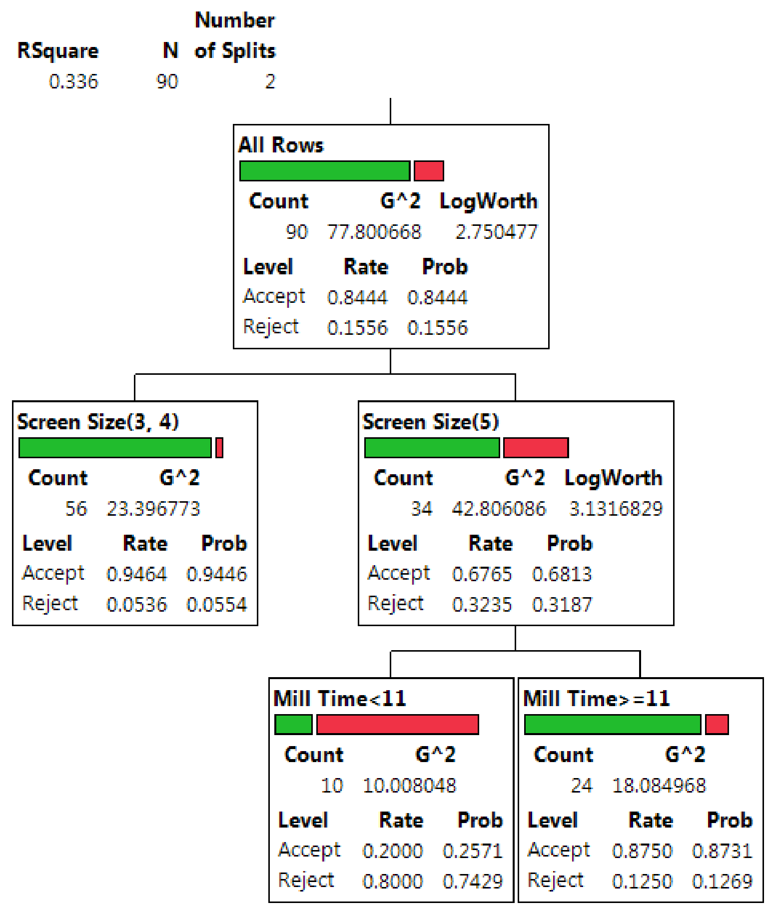
1st Split:

Optimal Split Screen Size 3 & 4 vs. Screen Size 5

Notice the difference in the rates in each branch of the tree

Repeat "Split Search" across both "Partitions" of the data. Find optimal split across both branches.

DECISION TREE
(STEP BY STEP)



2nd split on Mill Time
(< 11 vs. >= 11)

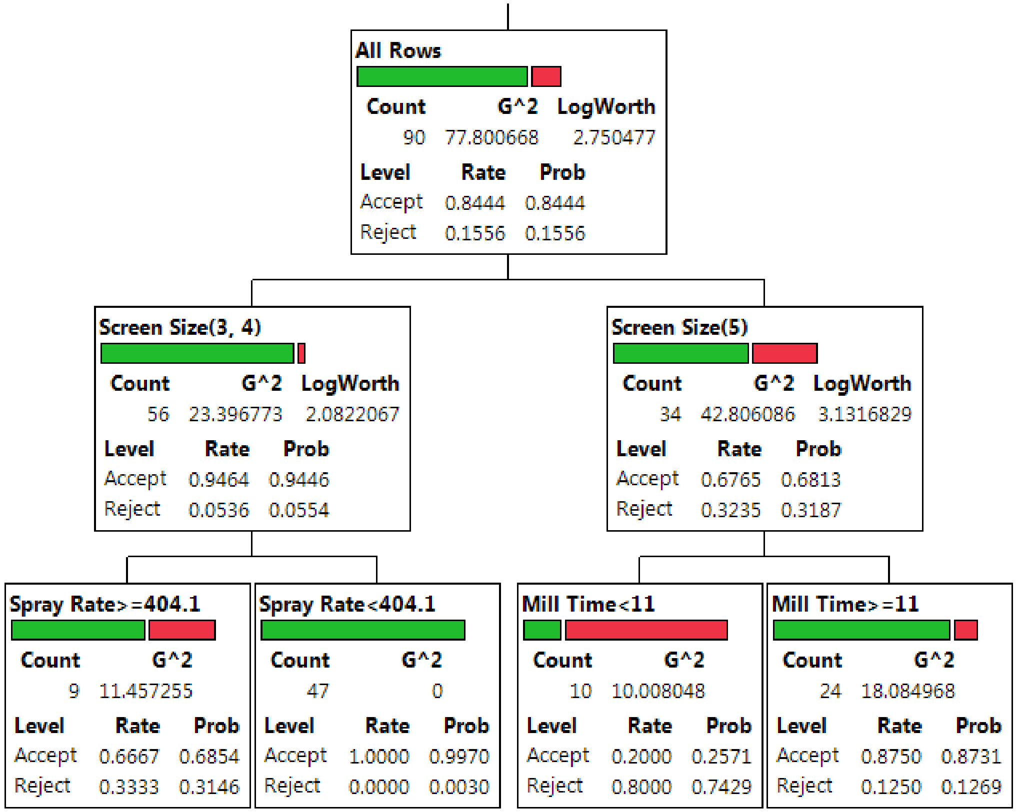
Notice variation in proportion of “1” in each branch

DECISION TREE (STEP BY STEP)

3rd split on Spray Rate (>= 404.1 vs. < 404.1))

Notice variation in proportion of “1” in each branch

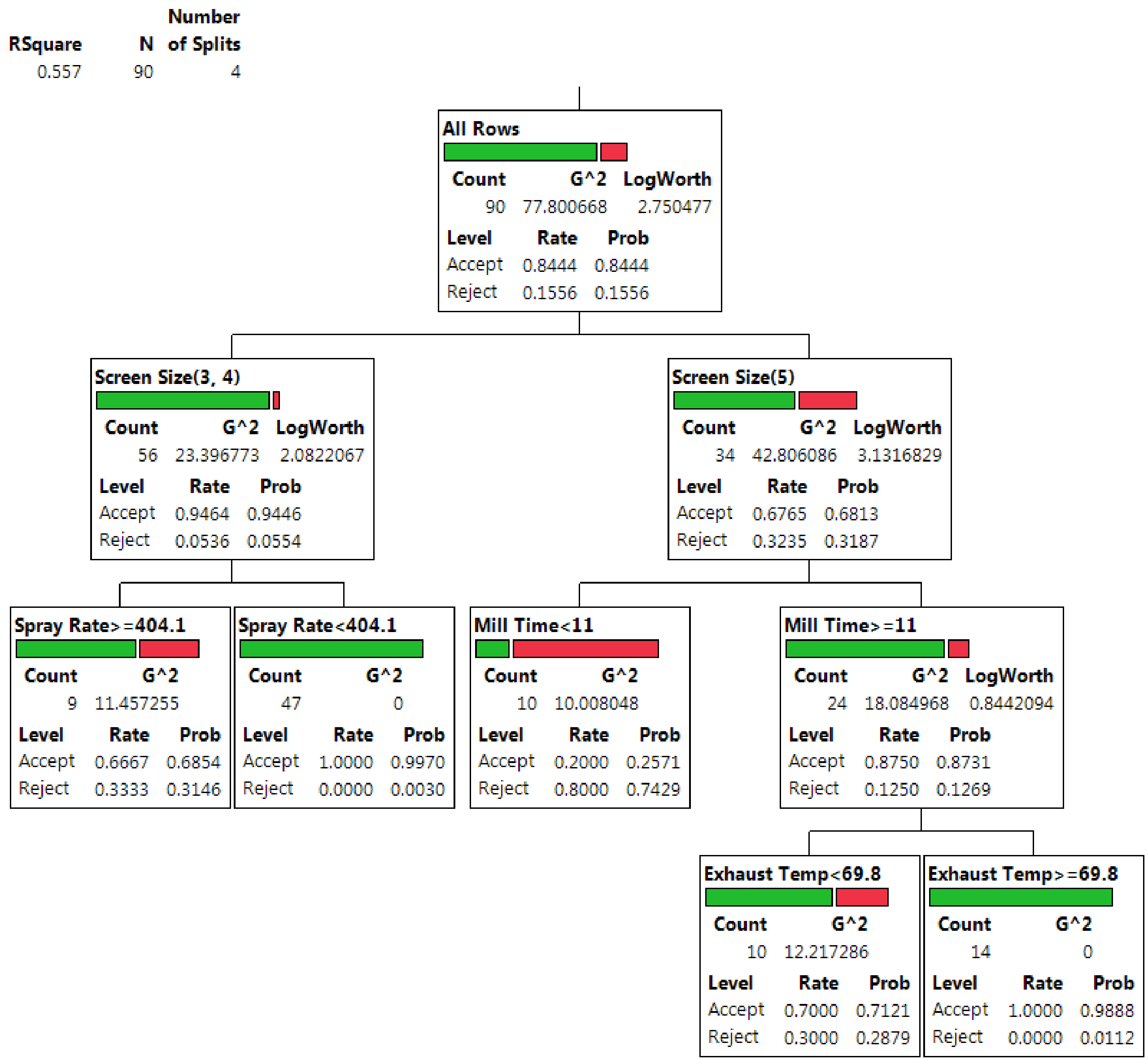
RSquare	N	Number of Splits
0.485	90	3



DECISION TREE (STEP BY STEP)

4th split on Exhaust Temp
(< 69.8 vs. >= 69.8)

Notice variation in proportion of “1” in each branch

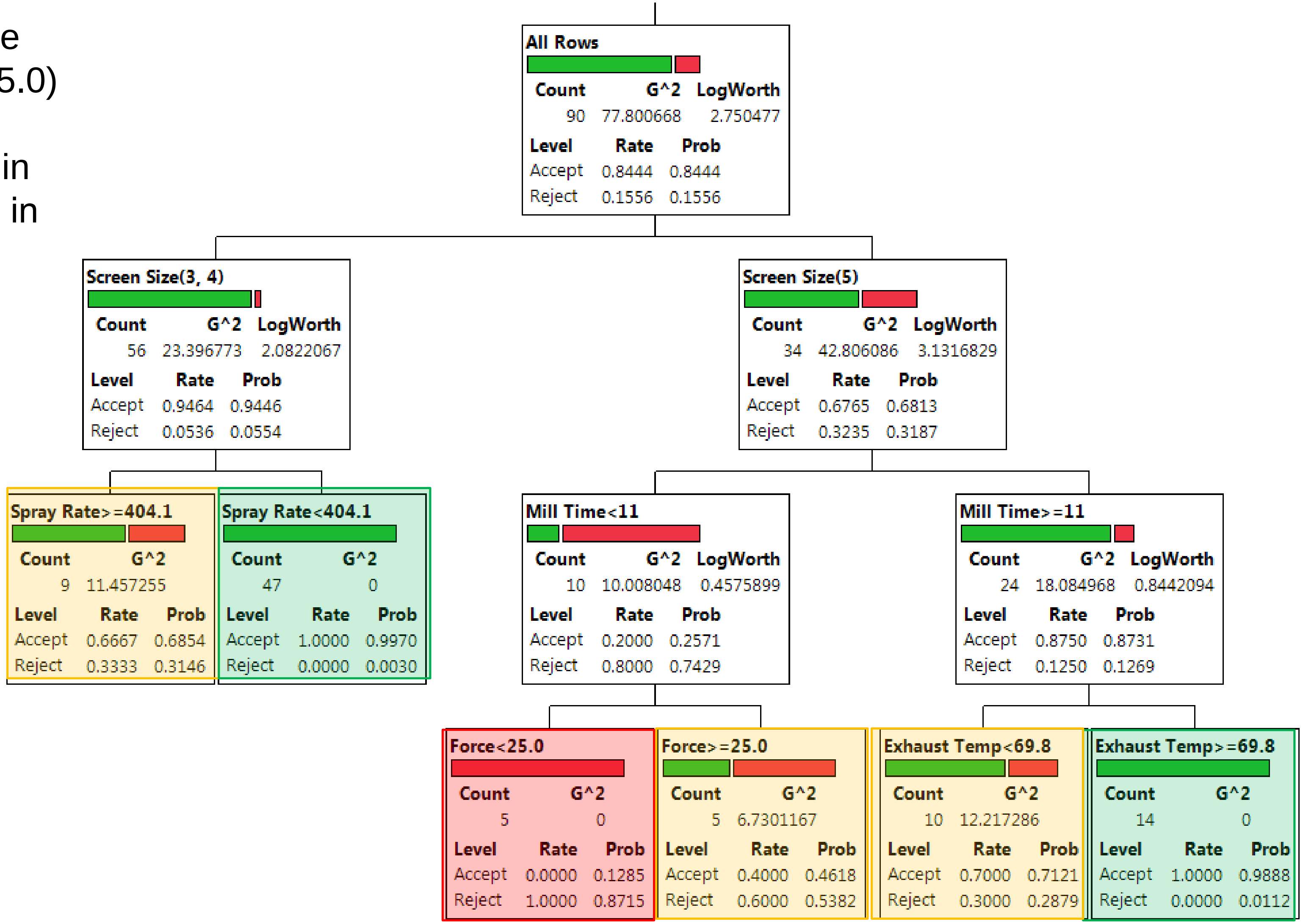


DECISION TREE (STEP BY STEP)

RSquare	N	Number of Splits
0.583	90	5

5th split on Force
(< 25.0 vs. >= 25.0)

Notice variation in proportion of “1” in each branch

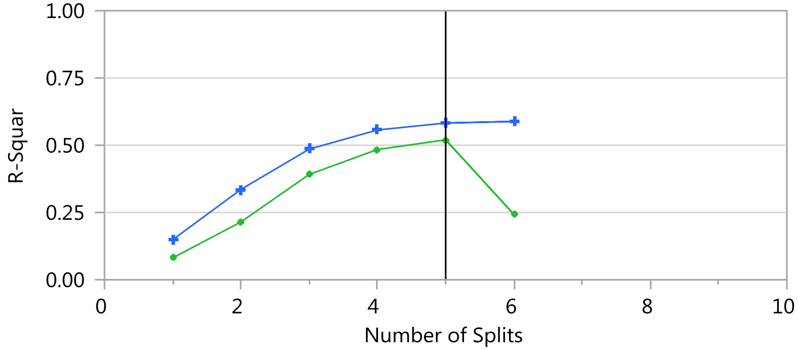


DECISION TREE (STEP BY STEP)

Crossvalidation

k-fold		-2LogLike	RSquare
5	Folde	37.3288048	0.5202
	Overa	30.4046577	0.5825

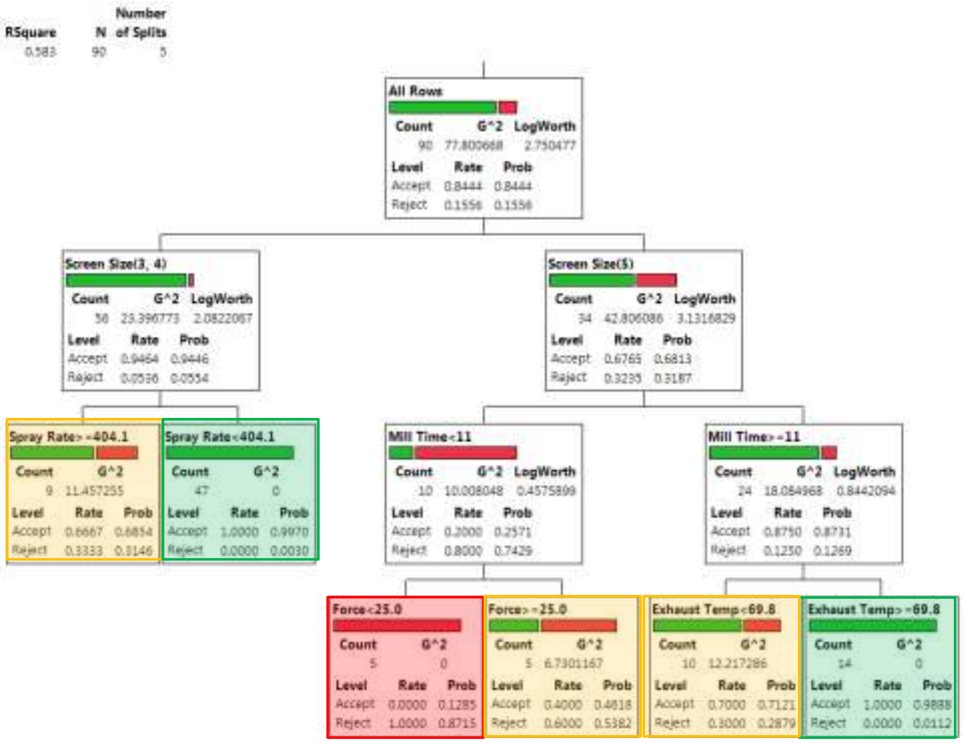
Split History



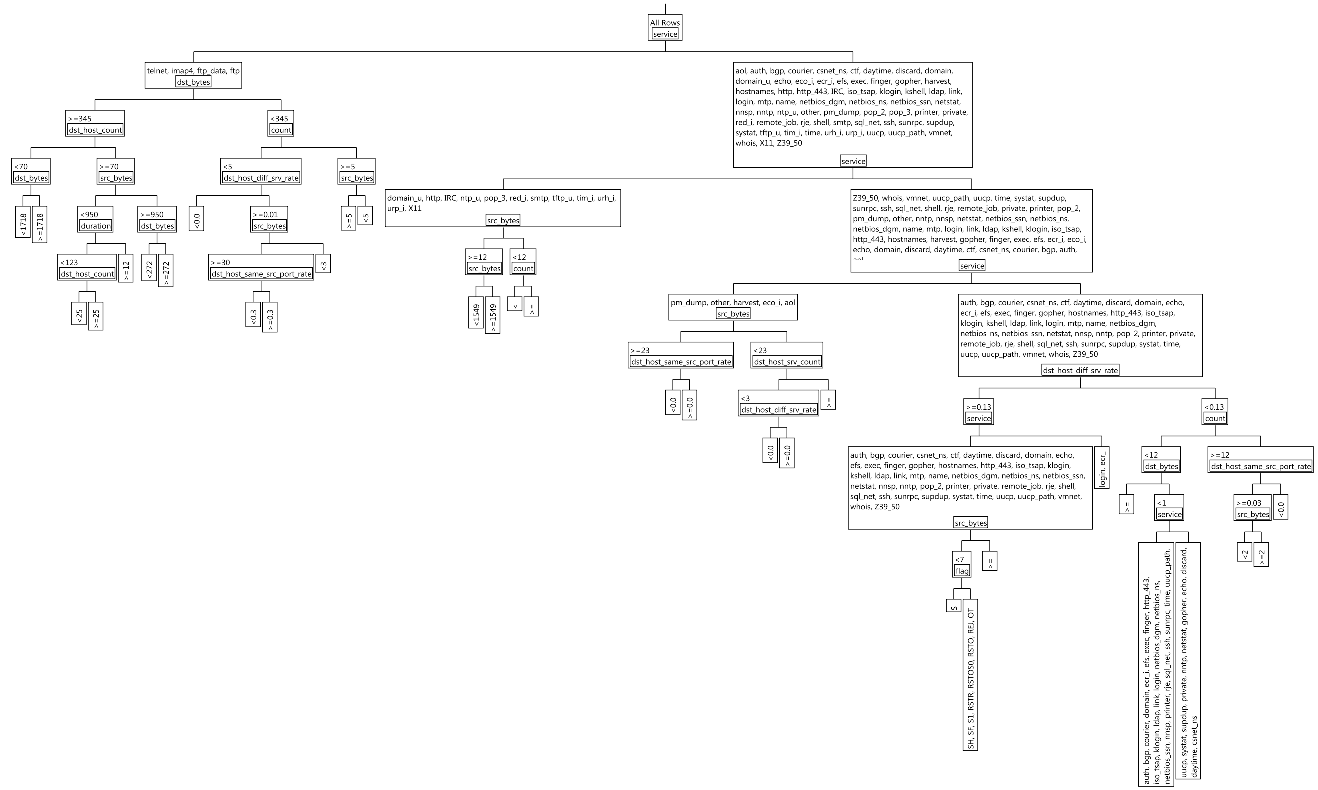
K-Fold in Green

Column Contributions

Term	Number of Splits	G^2	Portion
Mill Time	1	14.7130695	0.3104
Spray Rate	1	11.9395178	0.2519
Screen Size	1	11.5978092	0.2447
Exhaust Temp	1	5.8676817	0.1238
Force	1	3.2779318	0.0692
API Particle Size	0	0	0.0000
MgSt Supplier	0	0	0.0000
Lactose Supplier	0	0	0.0000
Sugar Supplier	0	0	0.0000
Talc Supplier	0	0	0.0000
Blend Time	0	0	0.0000
Blend Speed	0	0	0.0000
Compressor	0	0	0.0000
Coating Supplie	0	0	0.0000
Coating Viscosit	0	0	0.0000
Inlet Temp	0	0	0.0000
Atom. Pressure	0	0	0.0000

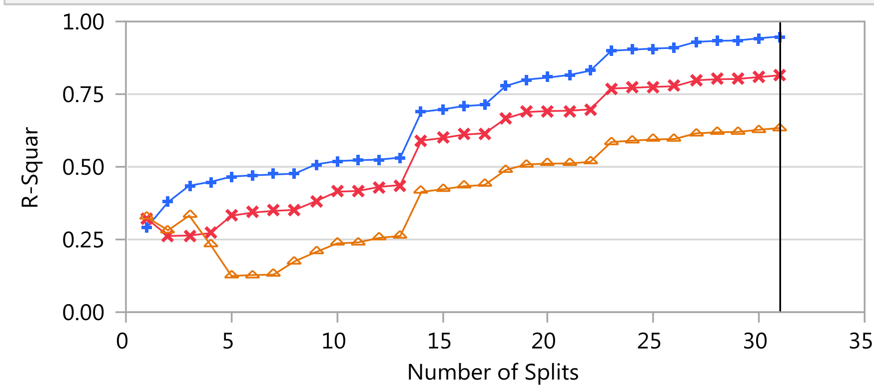


DECISION TREE 11 FACTORS



	RSquare	N	Number of Splits
Training	0.949	75582	31
Validatio	0.815	25194	
Test	0.634	25197	

Split History



Validation Data in Red

Test Data in Orange

Column Contributions

Term	Number of Splits	G^2	Portion
service	5	630992.402	0.5475
dst_bytes	4	128894.607	0.1118
dst_host_diff_srv_rate	3	115626.455	0.1003
src_bytes	8	97103.0428	0.0843
dst_host_count	2	71772.3696	0.0623
count	3	68716.3668	0.0596
dst_host_same_src_port_rat	3	19974.724	0.0173
dst_host_srv_count	1	10836.2482	0.0094
duration	1	5450.42578	0.0047
flag	1	3066.0292	0.0027
srv_count	0	0	0.0000

DECISION TREE - 11 FACTORS BOOTSTRAP FOREST

Measure	Training	Validation	Test
Entropy RSquare	0.9816	0.9798	0.9807
Generalized RSquar	0.9975	0.9972	0.9974
Mean -Log p	0.0296	0.0324	0.0312
RMSE	0.0834	0.0888	0.0868
Mean Abs Dev	0.0235	0.0253	0.0247
Misclassification Rat	0.0042	0.0055	0.0048

Column Contributions

Term	Number of Splits	G^2	Portion
service	313	6647269.76	0.3546
dst_bytes	318	2378144.67	0.1269
src_bytes	642	2343701.45	0.1250
dst_host_srv_count	545	1371395.91	0.0732
count	384	1361411.35	0.0726
dst_host_diff_srv_rate	435	988535.468	0.0527
flag	190	889445.342	0.0475
dst_host_same_src_port_rat	402	881707.319	0.0470
dst_host_count	435	700494.072	0.0374
srv_count	287	669775.801	0.0357
duration	222	511537.238	0.0273

DECISION TREE - 11 FACTORS

Measure	Training	Validation	Test
Entropy RSquare	0.9486	0.8149	0.6335
Generalized RSquar	0.9925	0.9661	0.9061
Mean -Log p	0.0828	0.2979	0.5898
RMSE	0.1426	0.2127	0.2811
Mean Abs Dev	0.0387	0.0637	0.0969
Misclassification Rat	0.0230	0.0495	0.0821

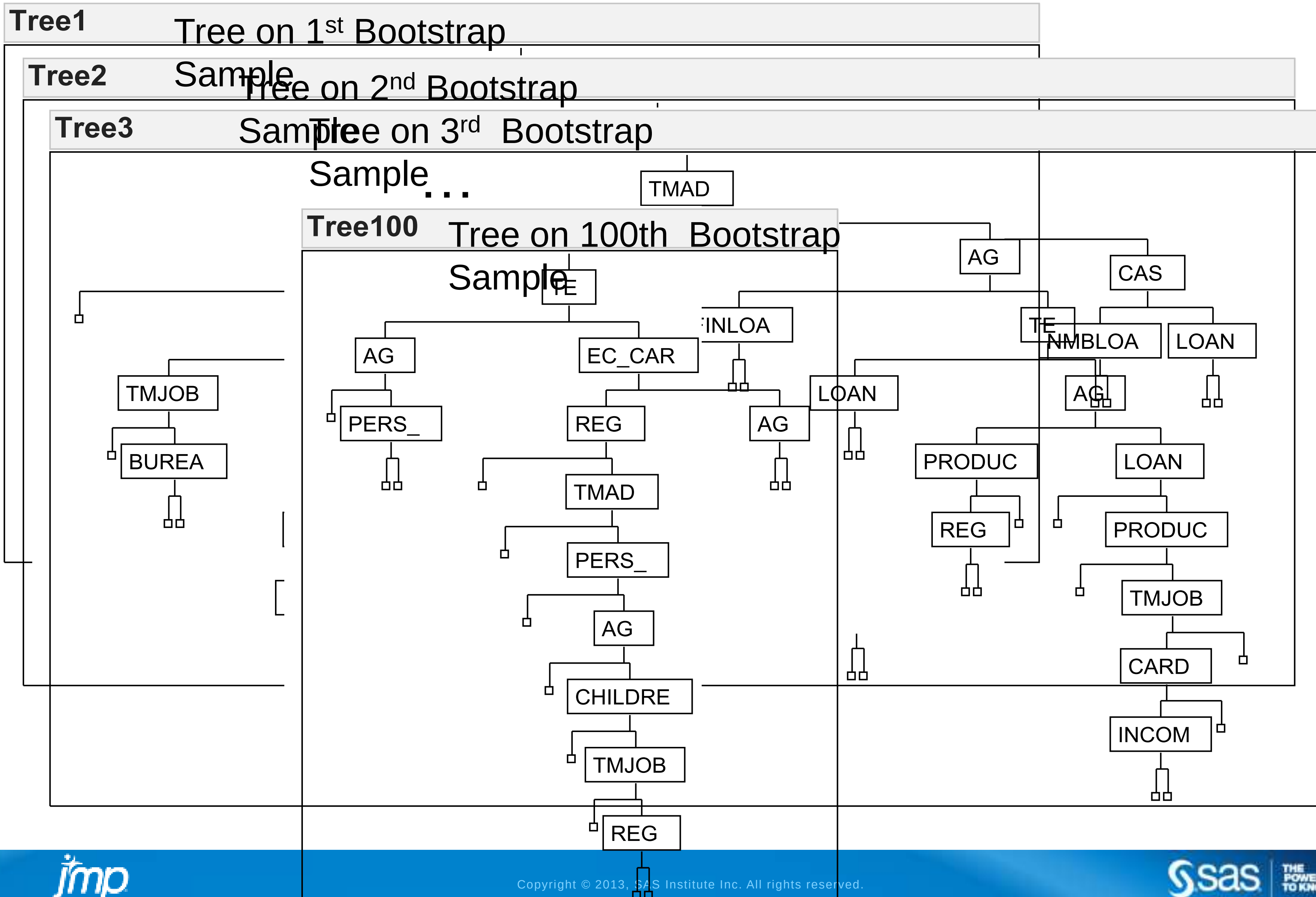
Column Contributions

Term	Number of Splits	G^2	Portion
service	5	630992.402	0.5475
dst_bytes	4	128894.607	0.1118
dst_host_diff_srv_rate	3	115626.455	0.1003
src_bytes	8	97103.0428	0.0843
dst_host_count	2	71772.3696	0.0623
count	3	68716.3668	0.0596
dst_host_same_src_port_rat	3	19974.724	0.0173
dst_host_srv_count	1	10836.2482	0.0094
duration	1	5450.42578	0.0047
flag	1	3066.0292	0.0027
srv_count	0	0	0.0000

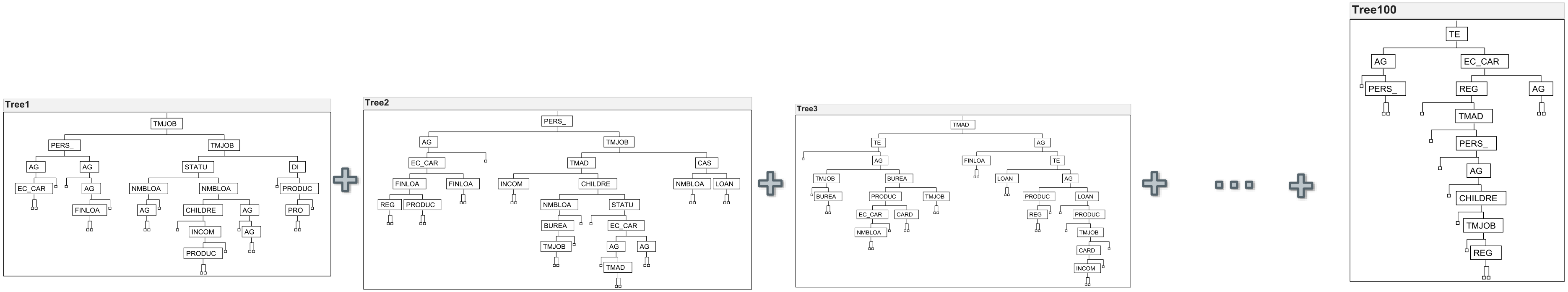
BOOTSTRAP FOREST

- Bootstrap Forest
  - For each tree, take a random sample of the predictor variables (***with replacement***) – e.g. pick half of the variables. Build out a decision tree on that subset of variables.
  - Make many trees and average their predictions (bagging)
  - This is also know as a random forest technique
  - Works very well on wide tables.
- Can be used for ***both*** predictive modeling and variable selection.
- Allows for dominant variables to be excluded from some trees giving less dominant – but still important – variables a chance to be selected.
- Valuable approach for screening variables for use with other modeling methods – e.g. neural networks.

SEE THE TREES IN THE FOREST



AVERAGE THE TREES IN THE FOREST



100

Bootstrap Forest Model

COLUMNS CONTRIBUTIONS – VARIABLE SELECTION W/44 FACTORS ORIGINAL 41 FACTORS + RANDOM (NORMAL, UNIFORM & INTEGER)

Column Contributions

Term	Number of Splits	G^2	Portion
service	450	10603400.8	0.2831
dst_bytes	382	5308498.33	0.1417
src_bytes	820	4771327.16	0.1274
count	337	2700247.28	0.0721
dst_host_srv_count	528	1990388.66	0.0531
dst_host_diff_srv_rate	415	1575488.06	0.0421
flag	168	1153015.42	0.0308
srv_count	238	1115688.05	0.0298
dst_host_serror_rate	175	1060259.19	0.0283
duration	276	991351.909	0.0265
dst_host_count	499	714300.159	0.0191
dst_host_same_src_port_rat	389	616742.634	0.0165
hot	159	535399.996	0.0143
same_srv_rate	103	422795.794	0.0113
dst_host_same_srv_rate	334	421699.768	0.0113
diff_srv_rate	145	382986.204	0.0102
serror_rate	65	365667.013	0.0098
dst_host_rerror_rate	233	318445.492	0.0085
dst_host_srv_serror_rate	117	308717.284	0.0082
logged_in	40	305603.637	0.0082
srv_serror_rate	30	219339.913	0.0059
root_shell	32	203921.266	0.0054
dst_host_srv_diff_host_rate	253	196905.011	0.0053
Random Uniform	228	195145.878	0.0052
dst_host_srv_rerror_rate	81	153228.513	0.0041
protocol_type	53	152857.046	0.0041
is_guest_login	12	137886.036	0.0037
Random Normal	194	110253.474	0.0029
num_compromised	39	76703.4706	0.0020
num_file_creations	20	75279.6937	0.0020
wrong_fragment	29	72313.7688	0.0019
rerror_rate	45	59525.1111	0.0016
num_root	23	41990.5367	0.0011
Random Integer	146	21117.3276	0.0006
srv_diff_host_rate	33	17448.0232	0.0005
num_failed_logins	7	17407.5895	0.0005
srv_rerror_rate	30	16080.2873	0.0004
num_access_files	11	11528.8834	0.0003
num_shells	11	8067.77994	0.0002
urgent	4	3131.15585	0.0001
su_attempted	1	42.7170189	0.0000
land	0	0	0.0000
num_outbound_cmds	0	0	0.0000
is_host_login	0	0	0.0000

Column Contributions

Term	Number of Splits	G^2	Portion
service	450	10603400.8	0.2831
dst_bytes	382	5308498.33	0.1417
src_bytes	820	4771327.16	0.1274
count	337	2700247.28	0.0721
dst_host_srv_count	528	1990388.66	0.0531
dst_host_diff_srv_rate	415	1575488.06	0.0421
flag	168	1153015.42	0.0308
srv_count	238	1115688.05	0.0298
dst_host_serror_rate	175	1060259.19	0.0283
duration	276	991351.909	0.0265
dst_host_count	499	714300.159	0.0191
dst_host_same_src_port_rat	389	616742.634	0.0165
hot	159	535399.996	0.0143
same_srv_rate	103	422795.794	0.0113
dst_host_same_srv_rate	334	421699.768	0.0113
diff_srv_rate	145	382986.204	0.0102

Top 11 of 44

Model Validation-Set Summaries

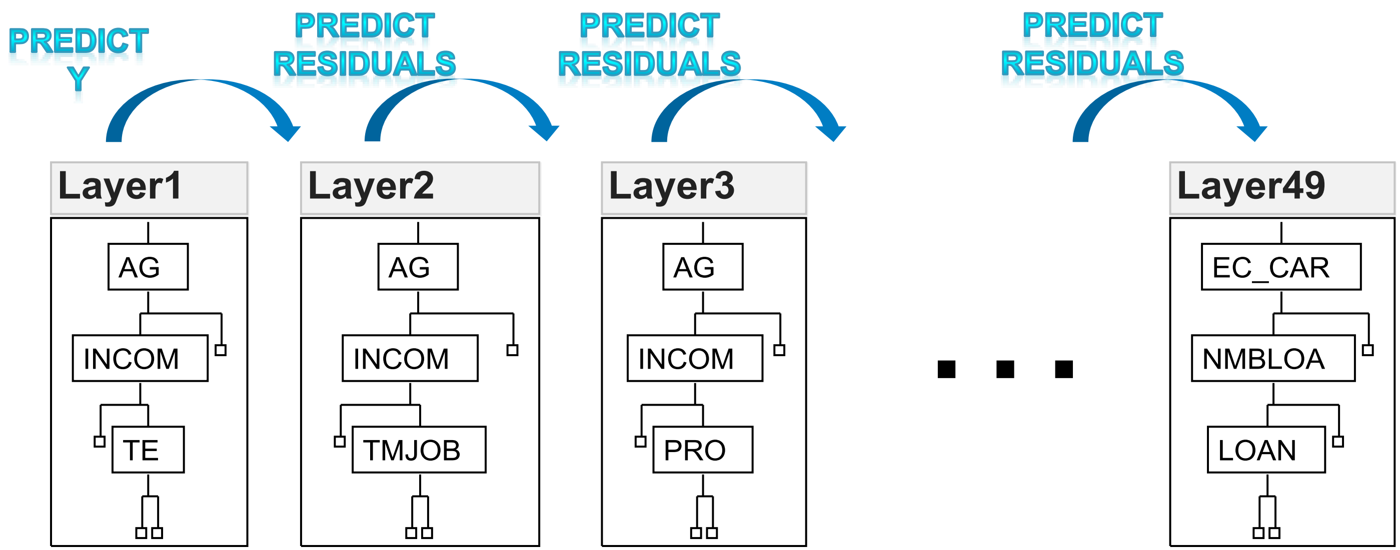
The fit below was the best of these models fit.

N Terms	N Trees	Entropy RSquare	Misclassification Rate	Avg -Log p	RMS Error	Avg Abs Error
11	200	0.9786	0.0040	0.0336	0.0856	0.0279
14	53	0.9811	0.0040	0.0297	0.0816	0.0243
18	48	0.9831	0.0039	0.0265	0.0770	0.0215

BOOSTED TREE

- Beginning with the first tree (layer) build a small simple tree.
- From the residuals of the first tree, build another small simple tree.
- This continues until a specified number of layers has been fit, or a determination has been made that adding successive layers doesn’t improve the fit of the model.
- The final model is the weighted accumulation of all of the model layers.

BOOSTED TREE ILLUSTRATED



Models M1 M2 M3 M49

Final Model

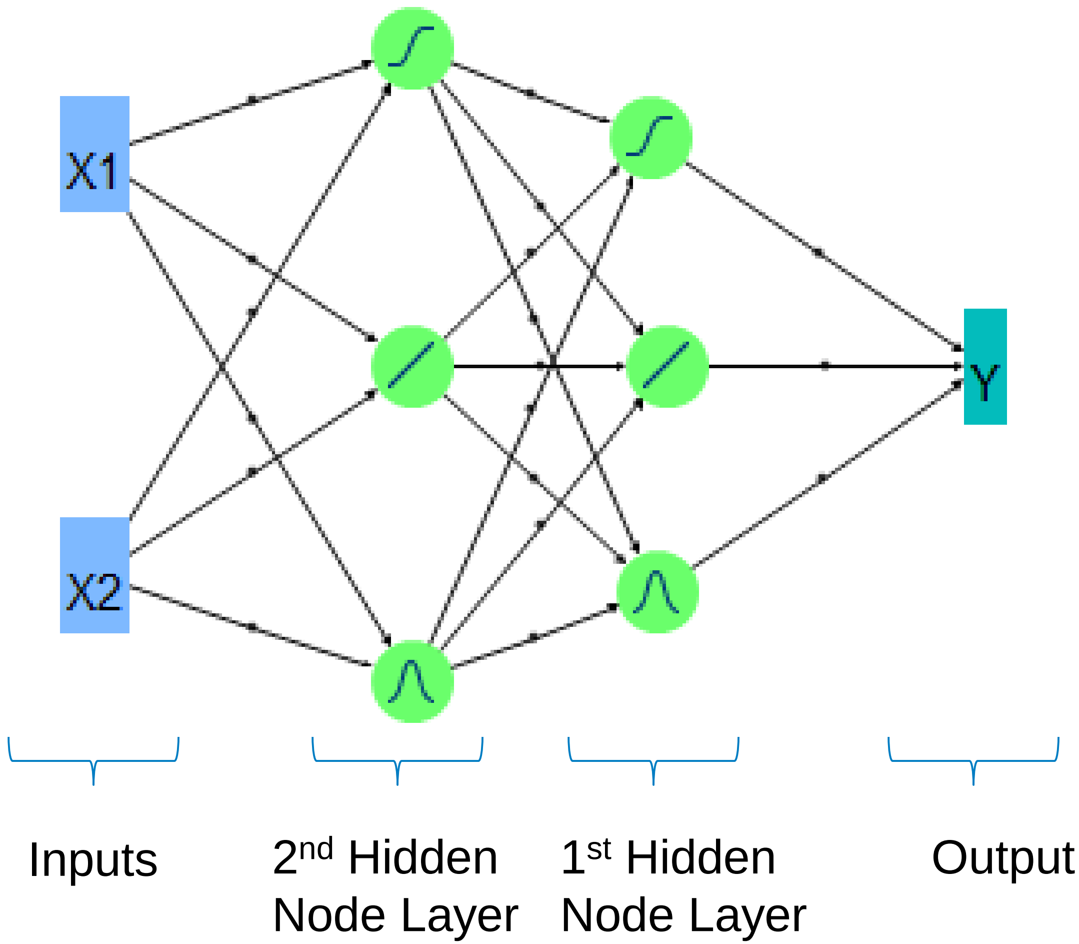
$$M = M1 + \varepsilon \cdot M2 + \varepsilon \cdot M3 + \cdots + \varepsilon \cdot M49$$

ε is the learning rate

NEURAL NETWORKS

- Neural Networks are highly flexible nonlinear models.
- A neural network can be viewed as a weighted sum of nonlinear functions applied to linear models.
  - The nonlinear functions are called activation functions. Each function is considered a (hidden) node.
  - The nonlinear functions are grouped in layers. There may be more than one layer.
- Consider a generic example where there is a response Y and two predictors $X1$ and $X2$. An example type of neural network that can be fit to this data is given in the diagram that follows

EXAMPLE NEURAL NETWORK DIAGRAM



NEURAL NETWORKS

- Big Picture
  - Can model:
    - Continuous and categorical predictors
    - Continuous and categorical responses
    - Multiple responses (simultaneously)
  - Can be numerically challenging and time consuming to fit
  - NN models are very prone to overfitting if you are not careful
    - There are several ways to help prevent overfitting
      - Some type of validation is required

NEURAL NET - 11 FACTORS SINGLE-LAYER

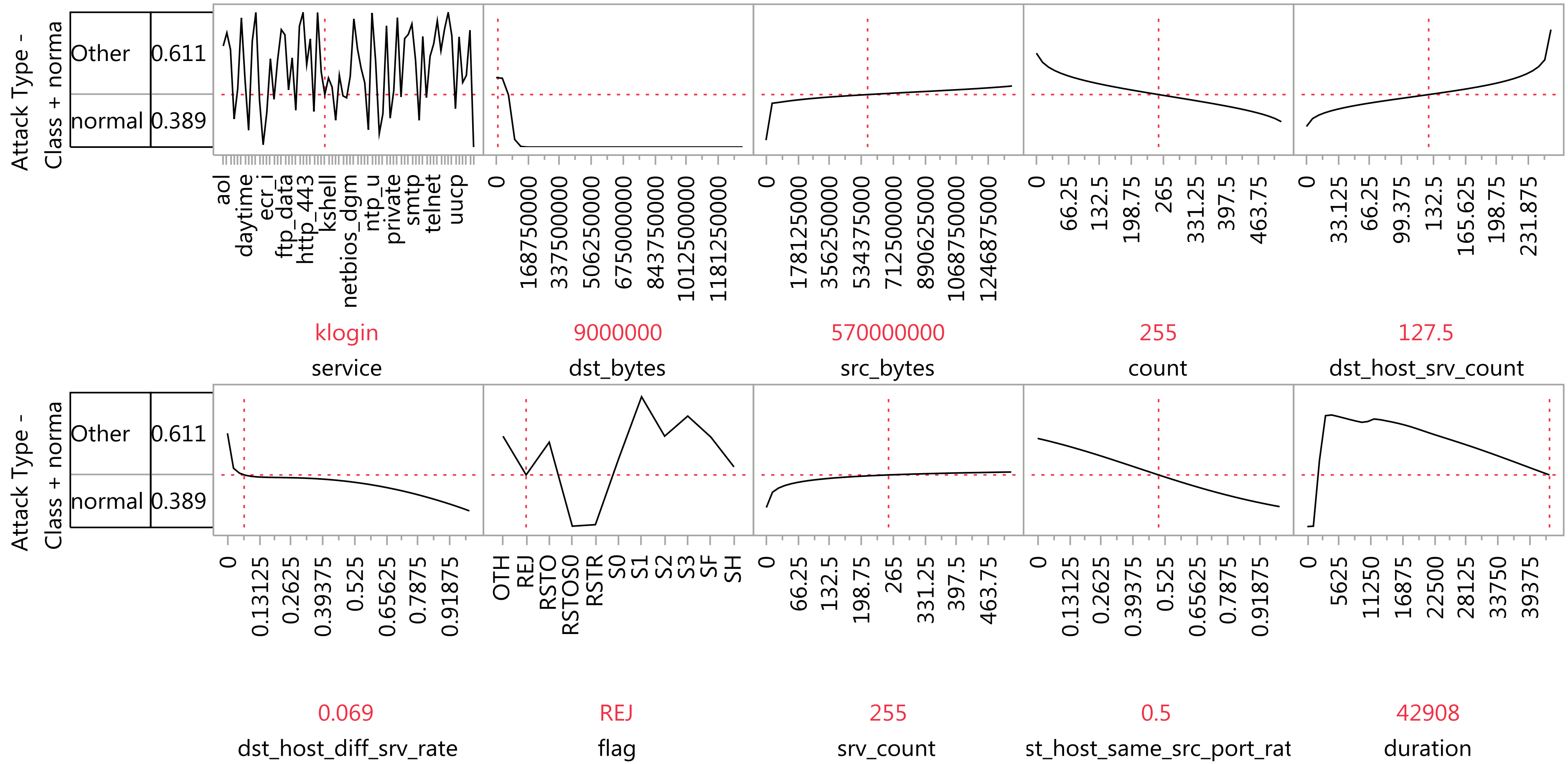
Measures	Value	Value	Value
Generalized RSquar	0.9814778	0.9764849	0.9805001
Entropy RSquare	0.8857414	0.8610009	0.8807691
RMSE	0.2171407	0.2374758	0.2165633
Mean Abs Dev	0.0928858	0.1022572	0.0937134
Misclassification Rat	0.0567399	0.0656212	0.0555819
-LogLikelihood	69405.509	27962.025	24450.71
Sum Freq	377425.96	125001.47	127437.57

NEURAL NET - 11 FACTORS BOOSTED

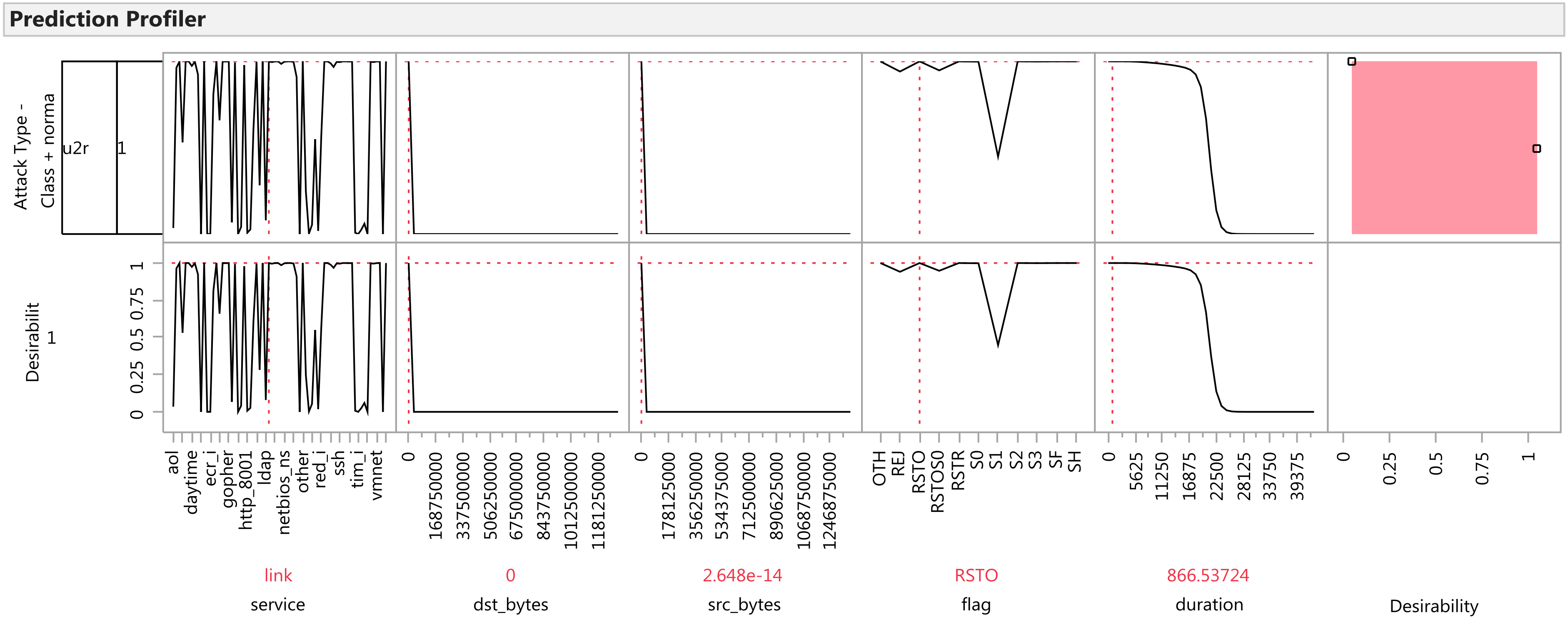
Measures	Value	Value	Value
Generalized RSquar	0.995034	0.9928519	0.9891299
Entropy RSquare	0.9650193	0.9508193	0.9280062
RMSE	0.11682	0.1384119	0.1827829
Mean Abs Dev	0.0364505	0.0414955	0.0632023
Misclassification Rat	0.0162761	0.0227104	0.0573684
-LogLikelihood	21248.789	9893.5268	14763.782
Sum Freq	377425.96	125001.47	127437.57

NEURAL MODEL PREDICTION PROFILER TOP 10 FACTORS

Prediction Profiler

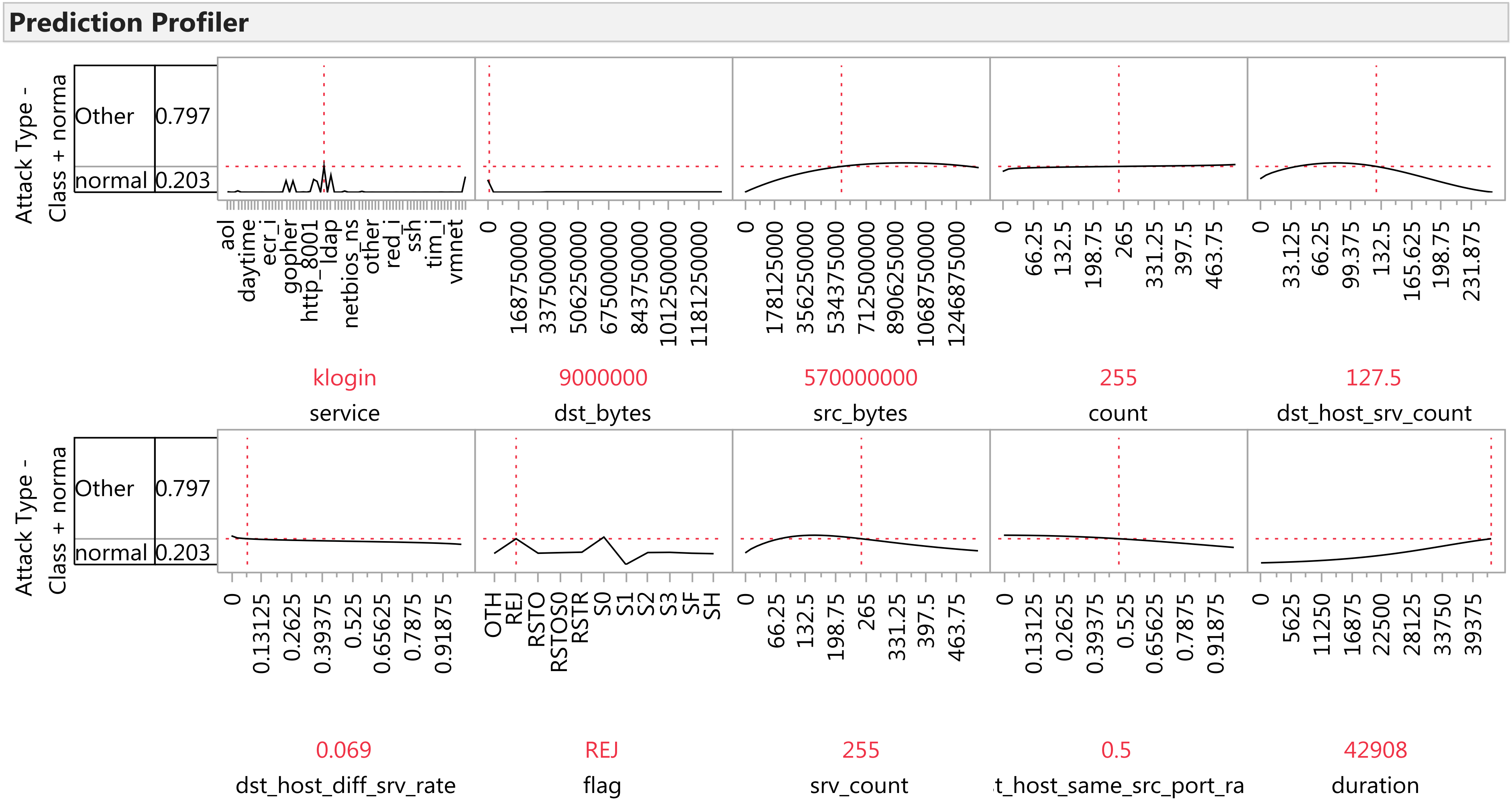


NEURAL MODEL PREDICTION PROFILER TOP 5 FACTORS



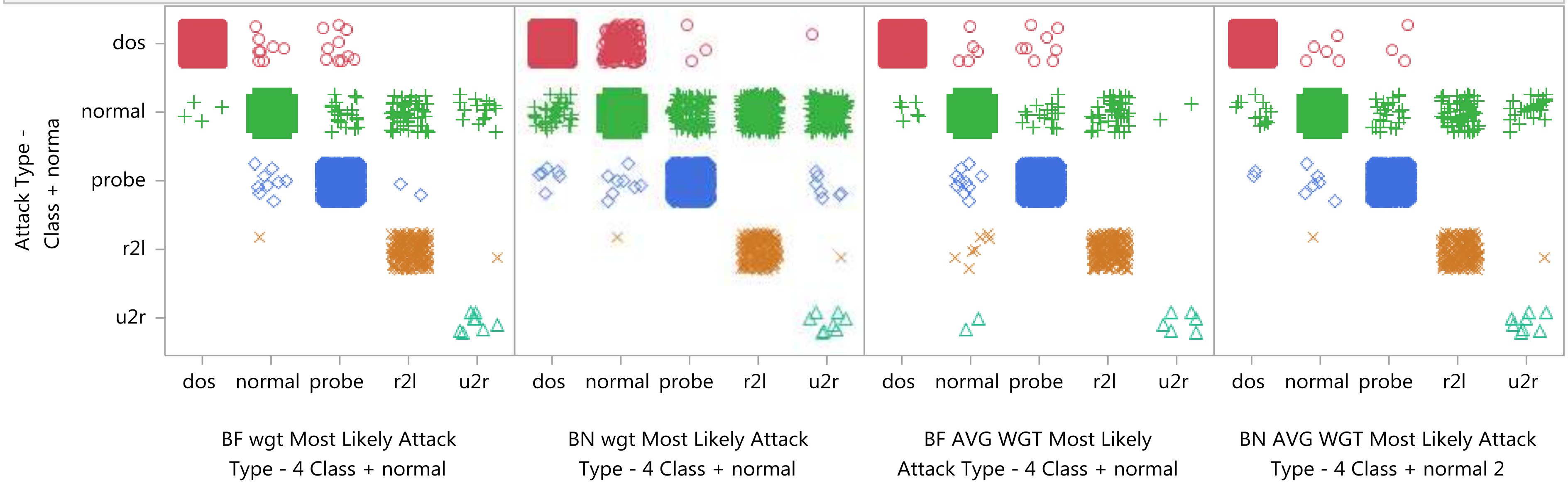
USE OPTIMIZATION TO FIND MOST PROBABLE CAUSE OF ATTACK TYPE

BOOTSTRAP FOREST PREDICTION PROFILER TOP 10 FACTORS

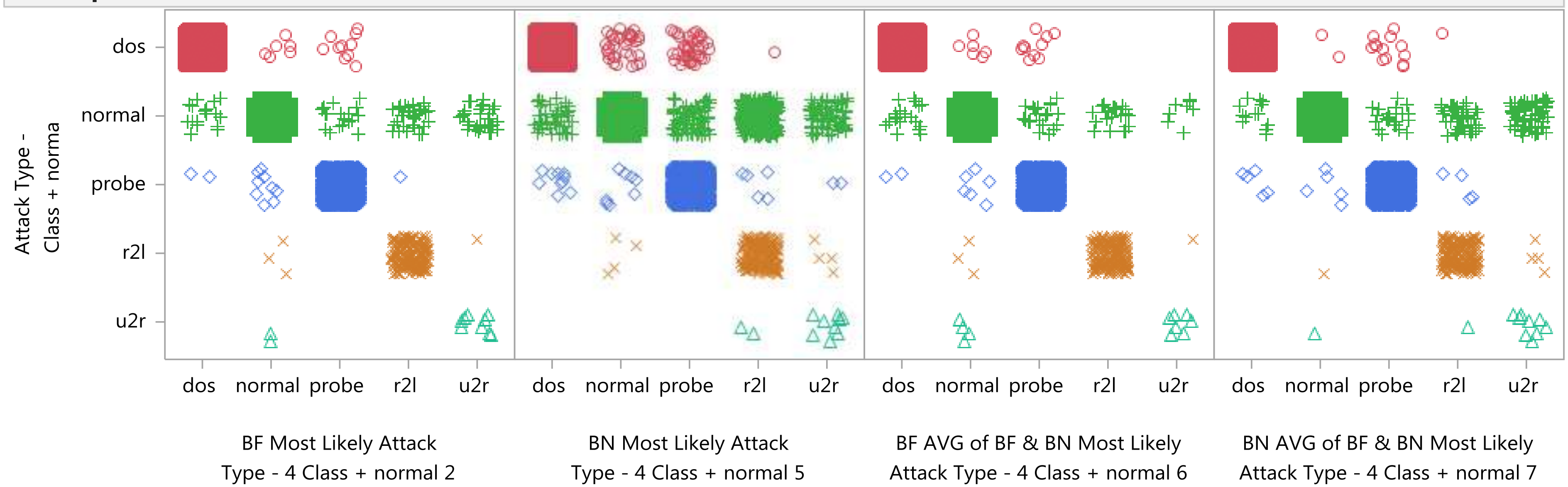


TOP – FIT 41 FACTORS | BOTTOM - FIT 11 FACTORS | RESULTS COMPARABLE

Scatterplot Matrix Holdback=2



Scatterplot Matrix TVT 60/20/20 Stratified=2



USE "NATE SILVER" APPROACH TO MODEL BIAS

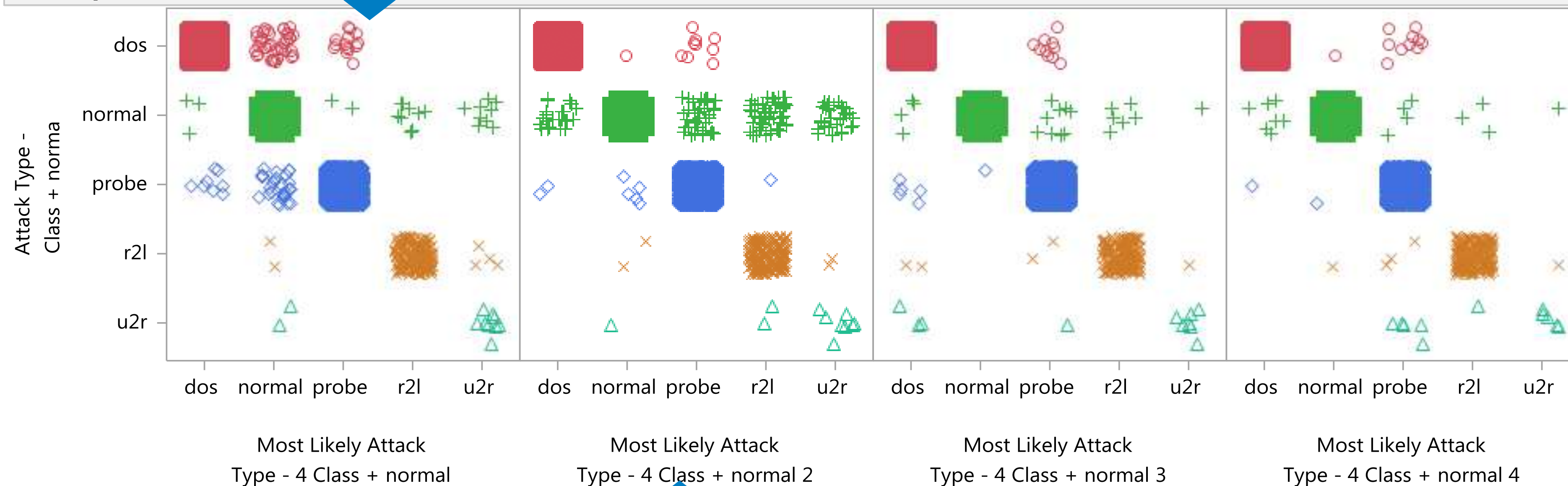
- Add a column of data that weights the misclassified cases differently than the correctly classified cases.
- More heavily penalize errors in predicting Normal than errors in predicting wrong Attacks
- If prediction worsens, then invert bias correction

ACTUAL VS. PREDICTED FOR TEST SUBSET FOR FOUR MODELS USING 11 FACTORS, ENSEMBLE MODELS AND BIAS

TED FOR

	TVT 60/20/20 Stratified				
	2				
	Most Likely Attack Type - 4 Class + norma				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9132	40	14	0	0
normal	3	13446	2	9	9
probe	8	29	2295	0	0
r2l	0	2	0	193	4
u2r	0	2	0	0	9

Scatterplot Matrix TVT 60/2 Stratified=2

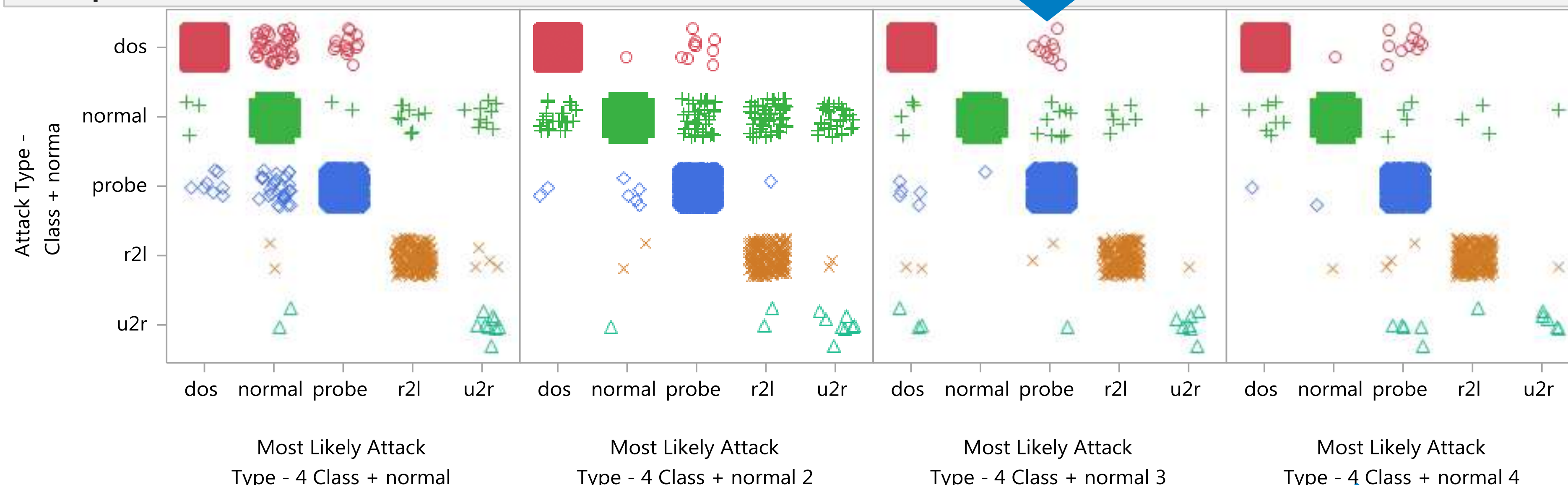


	TVT 60/20/20 Stratified				
	2				
	Most Likely Attack Type - 4 Class + normal				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9176	1	9	0	0
normal	24	13320	37	55	33
probe	2	5	2324	1	0
r2l	0	2	0	195	2
u2r	0	1	0	2	8

ACTUAL VS. PREDICTED FOR TEST SUBSET FOR FOUR MODELS USING 11 FACTORS, ENSEMBLE MODELS AND BIAS

	TVT 60/20/20 Stratified				
	2				
	Most Likely Attack Type - 4 Class + normal				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9176	0	10	0	0
normal	4	13448	10	6	1
probe	5	1	2326	0	0
r2l	2	0	2	194	1
u2r	3	0	1	0	7

Scatterplot Matrix TVT 60/20/20 Stratified=2



	TVT 60/2 [illegible] ratified				
	Most Likely Attack Type - 4 Class + normal				
Attack Type - 4 Class + normal	**dos**	**normal**	**probe**	**r2l**	**u2r**
dos	9175	1	10	0	0
normal	7	13454	4	3	1
probe	1	1	2330	0	0
r2l	0	1	3	194	1
u2r	0	0	5	1	5

HOW WOULD ONE USE THIS MODEL?

- Monitor factor settings by capturing 1 million rows of traffic
- Drop into proper columns as inputs
- Have model predict Attack Type
- If prediction is NOT Normal, then investigate further
- Repeat process and automate

IMPORTANT ISSUE

- Attackers are adaptive adversaries
- Must regularly update models

SUMMARY

- Fit several data mining models to historic cyber attack data
- Used Honest Assessment Approach of dividing data into Train, Validate and Test subsets to prevent overfitting of models
- Used “Ensemble” model averaging to improve prediction
- Used bias weighting of misclassified cases to further improve prediction

Thanks.
Questions or comments?

TOM.DONNELLY@JMP.COM