Identifying Important Features for Intrusion Detection

Using Support Vector Machines and Neural Networks

- **Objective**
- The dataset
- Ranking the significance of inputs
  - The Algorithms
  - Performance metrics for support vector machines
  - Performance metrics for neural networks
- Experiments
  - Experiments using support vector machines
  - Experiments using neural networks
- Summary & conclusions
- Comments

Objective

- Example models to detect intrusion
  - Support vector machine
  - Neural Network

- Simplify the model to make it faster and more accurate

How to simplify the model

- Elimination of the useless features
  - Rank the importance of input features
- Using a reduced number of features can deliver enhanced or comparable performance

- Objective
- **The dataset**
- Ranking the significance of inputs
  - The Algorithms
  - Performance metrics for support vector machines
  - Performance metrics for neural networks
- Experiments
  - Experiments using support vector machines
  - Experiments using neural networks
- Summary & conclusions
- Comments

Dataset

- In 1998 by Defense Advanced Research Projects Agency (DARPA)
- Originated from MIT's Lincoln Lab
- Raw TCP/IP dump
- The LAN was operated like a true environment
- Considered a benchmark for intrusion detection evaluations

Dataset

- Data size: 494021 examples
  - 20% of those examples are normal
- Number of features: 41
- Five possible classes
  - Normal
  - DOS: denial of service
  - R2L: unauthorized access from a remote machine
  - U2R: unauthorized access to local super user (root) privileges
  - Probing: surveillance and other probing

- Objective
- The dataset
- Ranking the significance of inputs
  - **The Algorithms**
  - Performance metrics for support vector machines
  - Performance metrics for neural networks
- Experiments
  - Experiments using support vector machines
  - Experiments using neural networks
- Summary & conclusions
- Comments

Approach

- Build the model and check the performance using all features
- Delete one feature at a time
- Build the model again, and verify the performance of the new model with the previous one.

The Algorithm

- Delete one input feature from the (training and testing) data
- Use the resultant data set for training and testing the classifier
- Analyze the results of the classifier, using the performance metrics
- Rank the importance of the feature according to the rules
- Repeat steps 1 to 4 for each of the input features

- Objective
- The dataset
- Ranking the significance of inputs
  - The Algorithms
  - **Performance metrics for support vector machines**
  - Performance metrics for neural networks
- Experiments
  - Experiments using support vector machines
  - Experiments using neural networks
- Summary & conclusions
- Comments

Performance metrics for support vector machines

- Ranks the importance of the 41 features in SVM-based IDS.
- Possible ranks for each feature
  - Important
  - Secondary
  - Insignificant

Performance metrics for support vector machines (SVM)

- Ranks based on three Performance criteria
  - Accuracy of classification
  - Training Time
  - Testing time
- There are total 10 possible rules for support vector machine

The rule set (SVM)

- ***If*** *accuracy* decreases ***and*** training time increases ***and*** testing time decreases, ***then*** the feature is important
- ***If*** *accuracy* decreases ***and*** training time increases ***and*** testing time increases, ***then*** the feature is important
- ***If*** *accuracy* decreases ***and*** training time decreases ***and*** testing time increases, ***then*** the feature is important

The rule set (SVM)

- ***If*** *accuracy* unchanges ***and*** training time increases ***and*** testing time increases, ***then*** the feature is important
- ***If*** *accuracy* unchanges ***and*** training time decreases *and* testing time increases, ***then*** the feature is secondary
- ***If*** *accuracy* unchanges ***and*** training time increases ***and*** testing time decreases, ***then*** the feature is secondary
- ***If*** *accuracy* unchanges ***and*** training time decreases ***and*** testing time decreases, ***then*** the feature is insignificant

The rule set (SVM)

- ***If*** *accuracy* increases ***and*** training time increases ***and*** testing time decreases, ***then*** the feature is secondary
- ***If*** *accuracy* increases ***and*** training time decreases ***and*** testing time increases, ***then*** the feature is secondary
- ***If*** *accuracy* increases ***and*** training time decreases ***and*** testing time decreases, ***then*** the feature is insignificant

- Objective
- The dataset
- Ranking the significance of inputs
  - The Algorithms
  - Performance metrics for support vector machines
  - **Performance metrics for neural networks**
- Experiments
  - Experiments using support vector machines
  - Experiments using neural networks
- Summary & conclusions
- Comments

Performance metrics for neural networks (NN)

- Three performance criteria
  - Overall accuracy (OA) of classification
  - False positive (FP) rate
  - False negative (FN) rate
- Possible ranks for the feature
  - Important
  - Secondary
  - Insignificant

The rule set (NN)

- *If* ***OA*** increases **and** ***FP*** decreases **and** ***FN*** decreases, then the feature is unimportant
- *If* ***OA*** increases **and** ***FP*** increases **and** ***FN*** decreases, then the feature is unimportant
- *If* ***OA*** decreases **and** ***FP*** increases **and** ***FN*** increases, then the feature is important
- *If* ***OA*** decreases **and** ***FP*** decreases **and** ***FN*** increases, then the feature is important
- *If* **OA** un-changes **and** ***FP*** un-changes, then the feature is secondary

Little introduction of the author & the paper

- The paper was published in 2003
- Number of citations until today is 493
- Srinivas Mukkamala
  - University of Southern Mississippi
  - Network security, computational intelligence
- Andrew H. Sung
  - New Mexico Institute of Mining and Technology
  - Machine learning, classification, Neural networks, pattern recognition

- Objective
- The dataset
- Ranking the significance of inputs
  - The Algorithms
  - Performance metrics for support vector machines
  - Performance metrics for neural networks
- Experiments
  - **Experiments using support vector machines**
  - Experiments using neural networks
- Summary & conclusions
- Comments

SVM characteristics

- SVM is a binary classifier
- Needs five models to classify the five classes
- Important features can be different for each model

Table1: Performance of SVMs using 41 features

Class	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	7.66	1.26	99.55
Probe	49.13	2.10	99.70
DOS	22.87	1.92	99.25
U2Su	3.38	1.05	99.87
R2L	11.54	1.02	99.78

Performance statistics with 40 features.

Table6: Class 1, Normal

Feature deleted	Training Time (sec)	Testing Time (sec)	Accuracy (%)
None	7.66	1.26	99.55
1.	10.19	1.11	99.51
2.	6.56	1.46	99.55
3.	9.06	1.47	99.48
4.	9.96	1.08	99.55
5.	33.11	1.62	99.19
6.	7.56	1.79	98.75
7.	7.11	1.43	99.55
8.	8.33	1.41	99.55
9.	8.37	1.37	99.55
10.	8.68	1.35	99.55
11.	7.49	1.33	99.55
12.	8.01	1.38	99.55
13.	7.14	0.81	99.55
14.	8.00	1.46	99.55
15.	9.81	1.43	99.55
16.	8.15	1.04	99.55
17.	8.12	1.47	99.55
18.	7.36	1.30	99.55
19.	8.00	1.12	99.55
20.	8.15	1.38	99.55
21.	7.98	1.42	99.55
22.	8.12	1.43	99.55
23.	7.65	1.34	99.56
24.	7.29	1.30	99.55
25.	8.32	1.35	99.55
26.	7.71	1.30	99.55
27.	7.73	1.38	99.55
28.	7.90	1.47	99.55
29.	7.81	1.39	99.55
30.	7.57	1.38	99.55
31.	7.11	1.30	99.55
32.	6.17	1.26	99.55
33.	8.53	1.51	99.48
34.	7.23	1.48	99.55
35.	6.96	1.35	99.55
36.	10.19	1.36	99.55
37.	6.74	1.33	99.55
38.	8.17	1.43	99.55
39.	7.75	1.32	99.55
40.	7.20	1.45	99.55

Table2: Performance of SVMs using important features

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	25	9.36	1.07	99.59
Probe	7	37.71	1.87	99.38
DOS	19	22.79	1.84	99.22
U2Su	8	2.56	0.85	99.87
R2L	6	8.76	0.73	99.78

Table3: Performance of SVMs using union of important features (30)

Class	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	7.67	1.02	99.51
Probe	44.38	2.07	99.67
DOS	18.64	1.41	99.22
U2Su	3.23	0.98	99.87
R2L	9.81	1.01	99.78

Table4: Performance of SVMs using important and secondary features

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	39	8.15	1.22	99.59
Probe	32	47.56	2.09	99.65
DOS	32	19.72	2.11	99.25
U2Su	25	2.72	0.92	99.87
R2L	37	8.25	1.25	99.80

- Objective
- The dataset
- Ranking the significance of inputs
  - The Algorithms
  - Performance metrics for support vector machines
  - Performance metrics for neural networks
- Experiments
  - Experiments using support vector machines
  - **Experiments using neural networks**
- Summary & conclusions
- Comments

Neural Network

- Consists of a collection of processing elements.
- Highly interconnected and transform a set of desired outputs.
- Multiple classes can be classified.
- Transformation is determined by the characteristics of the elements and the weights associated with the interconnections among them.

Delete features one by one (NN)

Table11: Neural network feature ranking results

Feature deleted	Accuracy (%)	False positive rate	False negative rate	Number of epochs
All	87.07	6.66	6.27	412
1	91.57	7.36	1.07	400
2	77.92	21.22	0.86	420
3	80.68	16.50	2.82	473
4	90.16	9.13	0.71	312
5	90.16	8.88	0.96	438
6	77.23	22.13	0.64	339
7	76.87	22.06	1.07	419
8	72.98	26.28	0.74	389
9	84.89	14.40	0.71	298
10	54.08	45.11	0.81	385

Table5: Neural network results using all 34 important features

No of features	Accuracy (%)	False positive rate	False negative rate	Number of epochs
41	87.07	6.66	6.27	412
34	81.57	18.19	0.25	27

- Objective
- The dataset
- Ranking the significance of inputs
  - The Algorithms
  - Performance metrics for support vector machines
  - Performance metrics for neural networks
- Experiments
  - Experiments using support vector machines
  - Experiments using neural networks
- **Summary & conclusions**
- Comments

Summary and Conclusions

- Important features gives the most remarkable performance in terms of training time.
- The most important features for the two classes of ‘Normal’ and ‘DOS’ heavily overlap
- ‘U2Su’ and ‘R2L’, the two smallest classes representing the most serious attacks, each has a small number of important features and a large number of secondary features

- Objective
- The dataset
- Ranking the significance of inputs
  - The Algorithms
  - Performance metrics for support vector machines
  - Performance metrics for neural networks
- Experiments
  - Experiments using support vector machines
  - Experiments using neural networks
- Summary & conclusions
- **Comments**

Comments

- Section: The rule set (SVM)
  - ***If*** *accuracy* decreases ***and*** training time *decreases* ***and*** testing time *decrease*, ***then*** the feature is …
  - ***If*** *accuracy* increases ***and*** training time increase ***and*** testing time increase, ***then*** the feature is …

- Section: The rule set (NN)
  - *If* **OA** un-changes **and *FP*** un-changes, then the feature is secondary.
    - Really! Doesn't make sense.

Comments

- Section: Delete features one by one (NN)
  - How to select which feature to remove is not clear.
  - The chart is not complete.
- Section: Summary and Conclusions
  - They claim something that we cannot verify with the existing information.
  - Contradicting conclusions:
    - Somewhere in conclusions, “The performances of using the important features do not show significant differences to that of using all 41 features.”
- Finally, It is a good concept of the elimination of unimportant features to make the more robust and faster.

Questions
&
Answers

Thank you!