# Machine Learning in Network Intrusion Detection

**Liru Long / Xilei Wang / Xiaoxi Zhu**
**CS229 Machine Learning**

## Data Pre-processiong

### Data Set

- Data set is downloaded from KDDCup'99 - annual Data Mining and Knowledge Discovery competition organized by ACM Special Interest Group on Knowledge Discovery and Data Mining
- Raw training/test data originates from TCP dump data of real network traffic, processed to generate seven million connection records.
- The subset used in this project includes a training set of 494021 samples and a test set with 311029 samples
- Each sample contains 41 features grouped into three categories: basic features, content features, and time-based features
- Each sample is labeled with integer 0~4 — 0 indicates normal traffic and 1~4 corresponds to a specific class of network attack as follows: 1—Probing, 2—DoS, 3—U2R, 4—R2L.

### Feature Pre-processing

Feature values in the original data set vary significantly ( continuous vs discrete, order of magnitude, etc). As many Machine Learning algorithms are sensitive to variations in feature values, sample features are normalized using following approaches:

1. Text features (e.g. protocol, service type) are mapped onto discrete integer values between 0 to 10
2. Large numerical values are normalized to the range [0, 1]
3. Extremely large numerical values (e.g. source bytes) are converted to range [1, 10] using logarithm base 10

## Naive Bayes Classifier

### Multi-class Classification

- A multinomial multi-class Naive Bayes Classifier is trained with training data set labeled with 0~4.
- Test data is classified into 0~4 with trained NB classifier.
- Incorrect labelling across different classes of attacks (e.g. label class-1 attack as class-2 attack) are ignored.

### Binary Classification

- A binary Naive Bayes classifier is trained with training data labeled as true vs false, (where label 0 mapped to true, labels 1~4 mapped to false).
- Test data is then classified as normal (true) vs attack (false).

***Key observation:***
- *Binary classifier & Multi-class classifier delivers similar performance*
- *Low precision of normal class posts potential security risk*

**Multi-class Classification**

| | Precision | Recall |
|---|---|---|
| Normal | 0.7164 | 0.941 |
| Attack | 0.9846 | 0.9098 |

**Binary Classification**

| | Precision | Recall |
|---|---|---|
| Normal | 0.7077 | 0.9557 |
| Attack | 0.9883 | 0.9045 |

*Precision vs Recall values for normal traffic vs attack*

### Protocol-based Classification

- Training data & test data are further divided into three subsets based on transport layer protocol - TCP, UDP and ICMP.
- Three multi-class NB classifiers are trained for each subset and test data is classified using corresponding classifier.

***Key observation:***
- *Improved performance in TCP-/ICMP-classifier*
- *Feature patterns of attack(e.g. DoS attack) correlated with protocol*

**TCP**

| | Precision | Recall |
|---|---|---|
| Normal | 0.7520 | 0.9918 |
| Attack | 0.9941 | 0.8082 |

**UDP**

| | Precision | Recall |
|---|---|---|
| Normal | 0.6056 | 0.9989 |
| Attack | 0.8882 | 0.0127 |

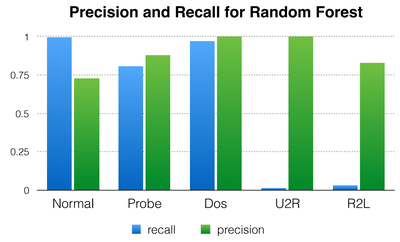
**ICMP**

| | Precision | Recall |
|---|---|---|
| Normal | 0.9583 | 0.1825 |
| Attack | 0.9981 | 0.9999 |

## Decision Tree & Random Forest

### Decision Tree

- By using the largest information gain, Decision Tree would build a tree based on the most distinguishable feature.
- Since decision tree suffered from overfitting, different max depths are tried. It turns out that depth of 9 gives the most satisfactory result.
- Decision Tree could successfully identify the normal network traffic with 98.4% recall, but with only 72.9% precision.
- Most U2R and R2L attacks are classified into normal class.




### Random Forest

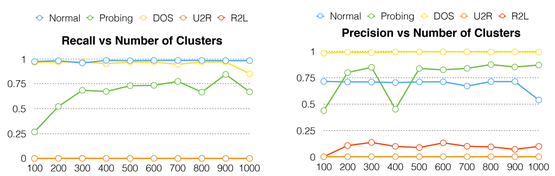
- Random Forest builds multiple decision tree using subsets of the training samples to improve classification or regression results.
- Since max depth of 9 is tried in decision tree method, decision trees in random forest all use max depth of 9 to build.
- Number of decision trees are varied when build the random forest; it turns out that with number of 16, we get the most satisfactory result.
- Random forest get every class improved, but for the attack and normal classification, it does not improve too much
- The random forest tends to classify the data with unseen pattern to normal class.

## K-means Clustering

- *Unsupervised learning:* suitable to deal with data with unknown characteristics
- *Time Complexity:* O(NKID)
- *Optimization:* Use mini-batch K-means to speed up the process

### Approach

- Normalize the data so that values of each feature range in [0, 1]
- Select the # of clusters k and generate random initial centroids for each cluster.
- The centroids are trained with training set.
- label each cluster by identifying the majority class of samples in this cluster




***Key Observations:***
- *Both precisions and recall improve significantly at small k (k < 300 for recall and k < 200 for precision), but improve slowly at larger k.*
- *Performance vary significantly due to random initialization of centroids. k = 300 generally gives the most stable result.*

## Feature Reduction

### Motivation

- The data set occupies extremely high-dimensional feature space
- Redundant features are identified through inspection based on prior networking knowledge (e.g. SYN error rate vs REJ error rate)
- Observation from single classifier results proves the assumption such that:
  - Features are correlated as overfitting has been detected - best performing DT classifier uses tree depth of 8 (against 41 features)
  - Certain features are more closely correlated with single class traffic - protocol based NB classifier generates improved performance

### Approach

- A pair-wise feature correlation matrix is generate
- Pairs of features with high correlation (pos & neg) are selected
- Identify groups of highly correlated features (each pair of features within the group are highly correlated)
- Apply PCA feature reduction to each group of highly correlated features.


### Results

The following group of features are identified as highly correlated:
- features 13 & 16
- features 2 & 23 & 24 & 36
- features 27 & 28 & 40 & 41
- features 4 & 5 & 29 & 33 & 34

Both training data & test data are reduced to 30-dimension after applying PCA

## Cascaded Classifier

### Cascaded DT-GMM

- The misclassification from R2L class to Normal class brings down the precision of normal class precision in all classifier models. The reason of this is lack of training samples.
- The Decision Tree Model gives the best result when classifying, using the Decision Tree Model as the first layer model.
- GMM model models the distribution of the data and does not suffer from initialization problem of K-Means.
- Unsupervised learning is preferred when dealing with unseen data.
- Building GMM to identify abnormal network traffic, after going through the decision tree model.
- The cascade improves the applications's ability to identify the attack but impairs its ability to classify normal traffic

**Cascaded Classifier**

| | Normal | Attack |
|---|---|---|
| Normal | 59110 | 1483 |
| Attack | 21286 | 229105 |

**Decision Tree**

| | Normal | Attack |
|---|---|---|
| Normal | 60251 | 342 |
| Attack | 22897 | 227539 |

Comparison of *Precision vs Recall values for normal traffic vs attack using single vs cascaded* classifier

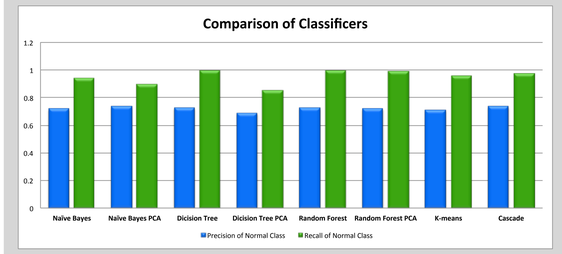
## Reflection & Future Work

### Results

- Consistent performance across different Machine Learning algorithms
- Decision Tree/Random forests outperforms the other algorithms with higher recall
- Precision in identifying normal traffic is **~70%**
- Recall of normal traffic classification vary across different approaches
- PCA shows marginal improvement in classification results
- Proposed extension - cascaded DT+GMM classifier - does not lead to expected significant performance improvement

### Reflections

- Dimension of data set features is in the medium to high range, which increases the complexity of learning
- As features are non-uniform - text vs numerical, range variation etc, results may be susceptible to data preprocessing
- Features are correlated - an effective feature reduction technique is critical to successful training
- Training data set and test data set do not have same sample distribution across multiple classes
- Sample size of class 3 (U2R attack) is too small
- Class 4 (R2L attack) samples in training data set and test data set do not have similar feature signature




### Future work

- A more effective data pre-processing technique may be developed, preserving unique feature characteristics
- Improved feature reduction technique - generate feature correlation for individual class to help identify signature patterns
- Finer classification of traffic types, especially for attack classes
- Explore unsupervised learning algorithm for effective abnormally detection as the network attack keeps evolving

## Reference


- *KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Web. Oct 2015.*
- *Scikit-learn tool, http://scikit-learn.org/stable/, web, Nov 2015*
- *M. Sabhnani and G. Serpen, 'Why Machine Learning Algorithms Fail in Misuse Detection on KDD Intrusion Detection Data Set', Intelligent Data Analysis, vol. 8, no. 1088-467, pp. 403-415, 2004.*
- *P. Gifty Jeya, M. Ravichandran and C. S. Ravichandran, 'Efficient Classifier for R2L and U2R Attacks - TechRepublic', TechRepublic, 2015.*
- *M. Sabhnani, 'Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context', In Proceedings of the International Conference on Machine Learning: Models, Technologies, and Applications, pp. 209-215, 2015.*