Research Paper Presentation - RP 12

Anirudh Srinivasan

IIT Hyderabad

CS20BTECH11059

Title and Authors

Title

A Performance Comparison of Data Mining Algorithms Based Intrusion Detection System for Smart Grid

Authors

1. Zakaria El Mrabet, School of Electrical Engineering & Computer Science, University of North Dakota, USA
2. Hassan El Ghazi, National Institute of Posts & Telecommunication, Rabat, Morocco
3. Naima Kaabouch, School of Electrical Engineering & Computer Science, University of North Dakota, USA

Abstract

1. Smart grid is an emerging and promising technology. It uses the power of information technologies to deliver intelligently the electrical power to customers.
2. Unfortunately, information technologies have inherent vulnerabilities and weaknesses that expose the smart grid to a wide variety of security risks.
3. Intrusion detection system (IDS) plays an important role in securing smart grid networks and detecting malicious activity, yet it suffers from several limitations.
4. This paper presents an overview of four data mining algorithms used by IDS in Smart Grid.

Key Concepts and Definitions

- **Smart Grid** : It is an electricity network enabling a two-way flow of electricity and data with digital communications technology enabling to detect, react and pro-act to changes in usage and multiple issues. Smart grids have self-healing capabilities and enable electricity customers to become active participants.
- **IDS** : It is a device or software application that monitors a network for malicious activity or policy violations.
- **Data Mining Algorithms** : These are a set of heuristics and calculations that creates a model from data. Optimal parameters for creating the mining model is found on analysing and these are then applied across the entire data set to extract actionable patterns and detailed statistics.

Introduction

1. Compared to the traditional power grid, smart grid provides new functionalities such as real-time control, two-way flow of information communication, operational efficiency, and grid resilience.
2. IDS is based on three distinct approaches in detecting abnormalities:
   - Signature-based: Detects patterns of malicious activities using a database of well-known attack signatures
   - Specification-based: Identifies deviation from normal behavior profiles using logical specifications
   - Anomaly-based: Looks for deviations from normal behavior profiles using statistical measures
3. The anomaly-based technique is more applicable and suitable approach for smart grid than signature-based and specification-based techniques.

Introduction Contd.

4. Despite its wide use, IDS suffers from several performance limitations, including limited detection accuracy and high rate of false positive.
5. This paper evaluates the performance of 4 Data mining algorithms used in IDS, which are:
   - Naïve Bayes
   - Decision tree
   - Support Vector Machine, and
   - Random Forest
6. The benchmark NSL-KDD (Network Security Laboratory - Knowledge Discovery in Databases) dataset is selected for this evaluation
7. The evaluation is based on the following metrics:
   - Probability of detection or true positive rate (TPR)
   - Probability of false alarm or false positive rate (FPR)
   - Probability of miss detection or false negative rate (FNR)
   - Efficiency and Processing time.

Naive Bayes

1. When the features/attributes of the data are independent, we can extend the Bayes Rule to what is called Naive Bayes, whose name is based on the naive assumption that the X's are independent of each other
2. **Bayes Rule:**

$$\Pr(Y|X) = \frac{\Pr(X|Y) \times \Pr(Y)}{\Pr(X)} \tag{1}$$

3. **Naive Bayes Rule:**

$$\Pr(Y = k|X_1 \ldots X_2) = \frac{\Pr(X_1|Y = k) \times \Pr(X_2|Y = k) \ldots \Pr(X_n|Y = k) \times \Pr(Y)}{\Pr(X_1) \times \Pr(X_2) \ldots \Pr(X_n)} \tag{2}$$

Naive Bayes Contd.

4. Let D be a training set of a tuple and their associated class labels. In our case, a tuple is a packet and it is represented by an n-dimensional attribute vector $X = \{x_1, x_2, x_3, \ldots, x_n\}$.
5. The number of classes = m = 2 (attack class, normal class). Given a tuple or a packet X, NB will predict that X belongs to the class with the highest posterior probability. In other words, X belongs to C_i if and only if:

$$\Pr(C_i|X) > \Pr(C_j|X) \text{ for } 1 \leq j \leq m, i \neq j \tag{3}$$

6.

$$\Pr(C_i) = \frac{|C_{i,D}|}{|D|} \tag{4}$$

where: $|C_{i,D}|$ is the number of training tuple of class C_i in D.

Naive Bayes Contd.

7. As NB assumes that there is no relationship among child nodes or attributes, the $\Pr(X|Ci)$ is given by:

$$\Pr(X|C_i) = \Pr(x_1|C_i) \times \Pr(x_2|C_i) \ldots \Pr(x_n|C_i) \tag{5}$$

where: x_k is the value of attribute A_k for tuple X.

8. As the attributes $A_1, A_2, \ldots A_n$ in the data set are categorical, we have:

$$\Pr(x_k|C_i) = \frac{|C_{i,D,x_k}|}{|C_{i,D}|} \tag{6}$$

where: $|C_{i,D,x_k}|$ is the number of tuples of class C_i in D having the value x_k for A_k

Decision Tree Algorithm

1. The decision tree is a supervised learning algorithm that can be used for addressing the classification problem. It consists of root node, branches nodes, and leaf nodes.
2. Each node represents a feature or attribute. Each link or branch represent a decision or rule and the leaf node is the class label to which a given object belongs.
3. **Entropy:** It is a measure of uncertainity in the dataset

$$S(D) = \frac{-p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right) \tag{7}$$

where: p is the number of positive examples and n is the number of negative examples in the dataset.

4. **Average information entropy:**

$$I(A_k) = \sum_{\forall i} \frac{p_i + n_i}{p+n} \times S(A_k = x_i) \tag{8}$$

Decision Tree Algorithm Contd

5. **Information Gain:** It is a measure of how much information the answer to a specific question provides. It is the difference in entropy before and after splitting the dataset on attribute A.

$$G(A_k) = S(D) - I(A_k) \tag{9}$$

6. The attribute that best classifies the training data should be used as the root node of the tree.
7. **Algorithm:**
   - Compute the entropy of the entire dataset
   - For every attribute or feature:
     - ★ Calculate entropy for all the values that the particular attribute can take
     - ★ Calculate average information entropy for the current attribute
     - ★ Calculate gain for the current attribute
   - Pick the attribute with highest gain for this node
   - Repeat until we get a complete decision tree with leaf nodes at the end.

Support Vector Machine

1. SVM is a machine learning algorithm used for classification and regression. It is based on a hyper line classifier, which separates and maximizes the margin between two classes
2. Let the data set D be given as $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)\}$ where x_i is the set of training tuples with associated class label y_i. Each y_i can take one of two values, either $+1$ or -1, corresponding to the class 'attack' and class 'normal'.
3. SVM finds the best decision boundary to separate two classes by searching for the maximum margin hyper line. A separating hyper line can be written as:

$$h(x) = \omega^T X + b \tag{10}$$

Support Vector Machine Contd.

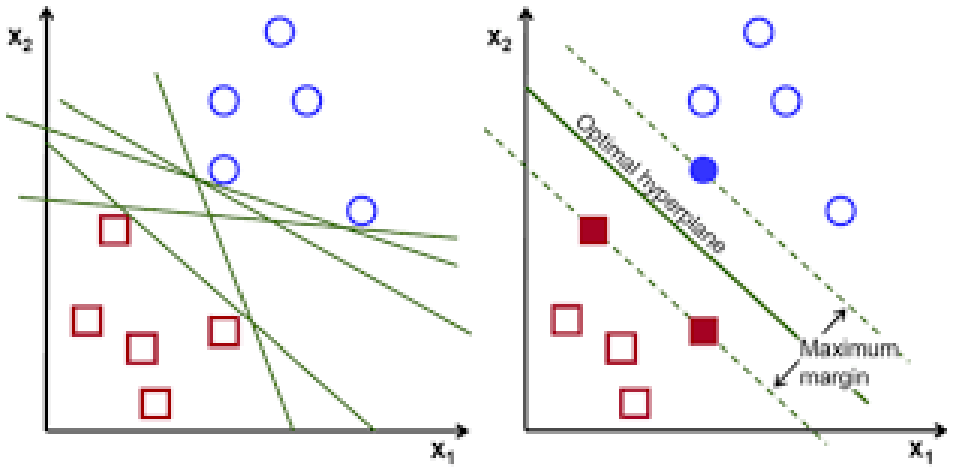

Figure: Diagram illustrating Support Vector Machines

4. For all the red colored points, h(x) is positive and is taken to be 1 after normalization, since it is a classification problem. Similarly, for all blue coloured points h(x) is negative and it taken to be -1.

Support Vector Machine Contd.

5. To find the optimum hyperplane, we have to find (ω,b) so as to:

$$\text{Maximize } \frac{2}{||\omega||} \text{ such that } y_i \times (\omega^T X + b) \geq 1 \tag{11}$$

6. In order to create a generalised model, the cost function for this algorithm can be given by:

$$J(\omega, b) = \frac{||\omega||}{2} + c_i \sum_{i=1}^{n} \epsilon_i \tag{12}$$

where: c_i = Number of permissible errors, which is also called regularization parameter and ϵ_i = value of errors

SVM Kernels

1. In the case of non-linear data, we can first transform the data through non-linear mapping to another higher dimension space and then use a linear model to separate the data. The mapping function is done by a kernel function

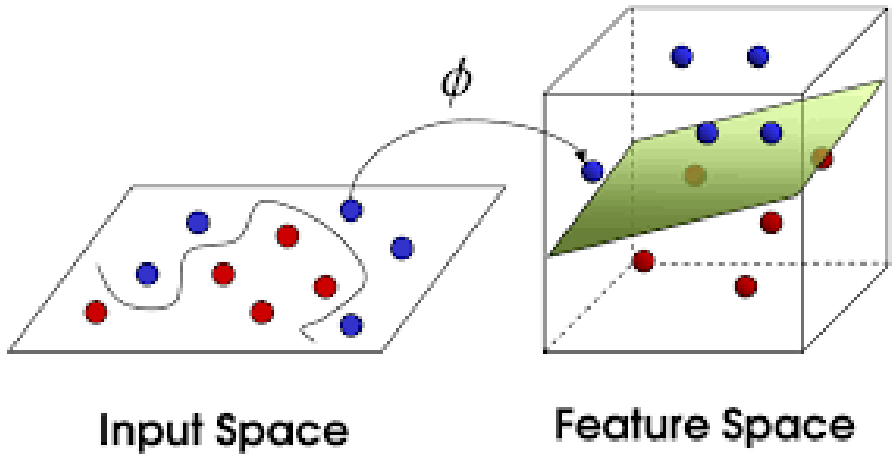

Figure: Diagram illustrating SVM Kernels

Random Forest

1. RF is a classifier which induces an ensemble of randomized decision trees for classification and prediction. Randomness is achieved either by selecting a sample from the training data set or by selecting randomly a subset of attribute at each node of each decision tree.
2. It has low bias and low variance unlike a normal decision tree algorithm which has low bias and high variance while predicting the class of test data.

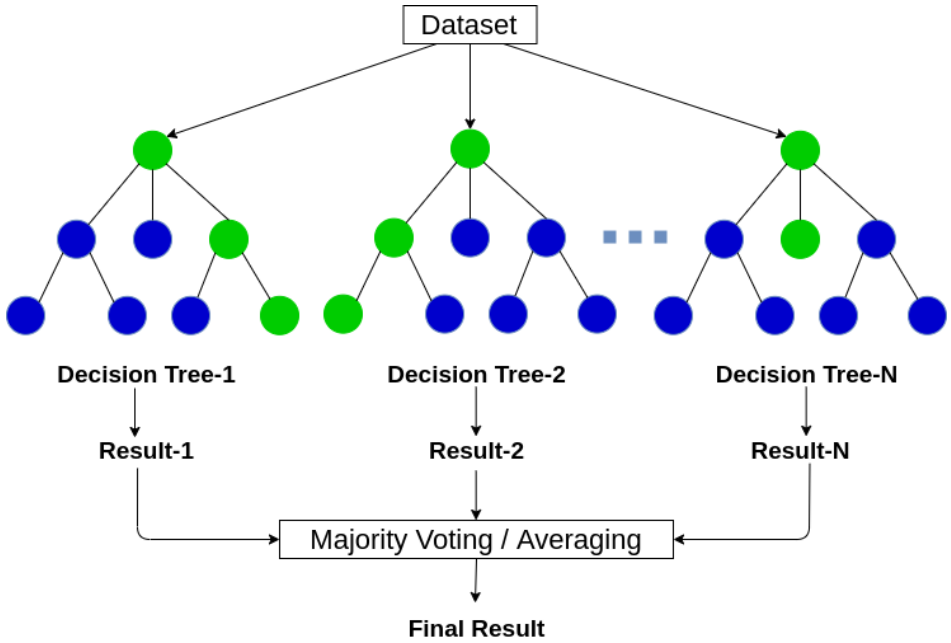

Figure: Diagram illustrating Random Forest Classifier

NSL-KDD Dataset

1. Although KDD Cup 99 is considered as a benchmark data set for assessing anomaly detection algorithms, it has a large number of redundant connections, especially a repeating record of DOS category.
2. In addition, there is a bias distribution of the four categories which makes an accurate classification of the U2R and R2L difficult. These issues have been addressed in the NSL-KDD data set.

Attack's category	**Description**	**Attack's Examples**
Remote to Local (R2L)	Unauthorized access from a remote machine	Password guessing
User to Root (U2R)	Unauthorized access to local root privileges from a local unprivileged user	Rootkits, buffer overflow attack
DOS	Denial of service	Teardrop attack and Smurf attack
Probing	Surveillance and scanning	Scanning attack

Table: Attack's categories in KDD CUP 99 and NSL-KDD

Comparison of the 4 Algorithms

1. The probability of detection or true positive rate (TPR) is given by:

$$TPR = \frac{TP}{TP + FN} \times 100 \tag{13}$$

2. The probability of false alarm or false positive rate (FPR) is given by:

$$FPR = \frac{FP}{FP + TN} \times 100 \tag{14}$$

3. The probability of miss detection or false negative rate (FNR) is given by:

$$FNR = \frac{FN}{TP + FN} \times 100 \tag{15}$$

4. The efficiency or accuracy is given by:

$$\text{Efficiency} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{16}$$

where: TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative

Results

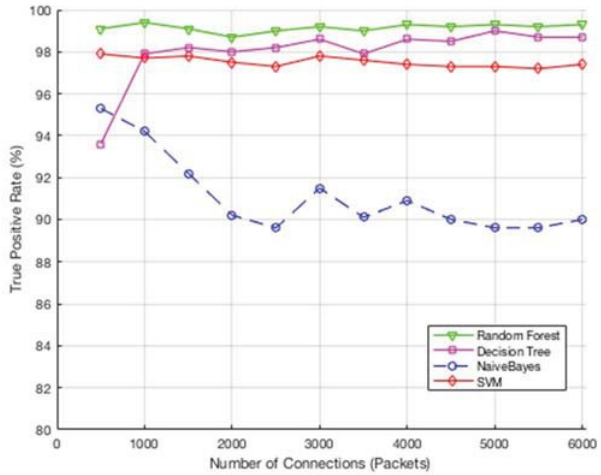

Figure: TPR vs Number of Packets

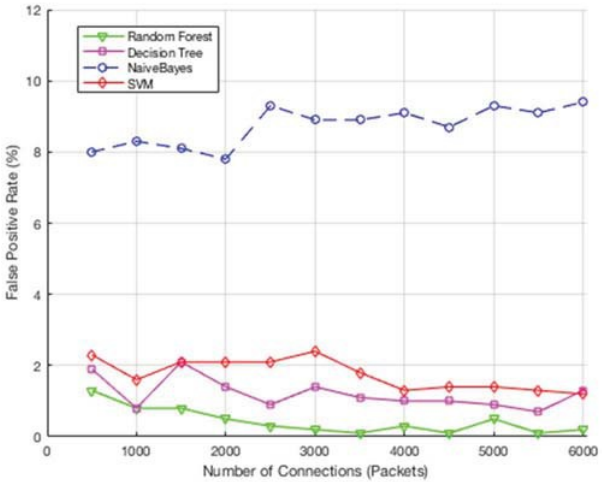

Figure: FPR vs Number of Packets

Results Contd.

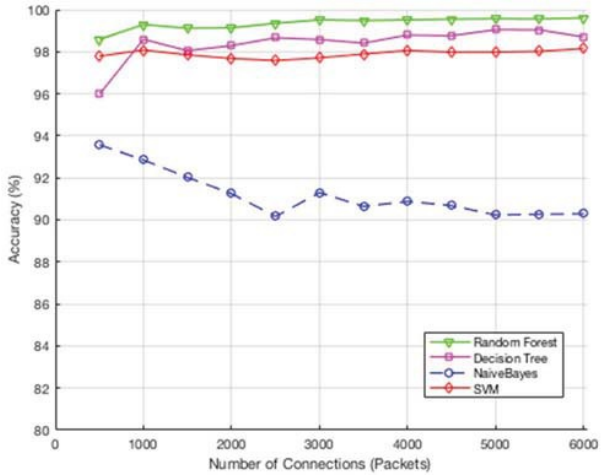

Figure: Accuracy vs Number of Packets

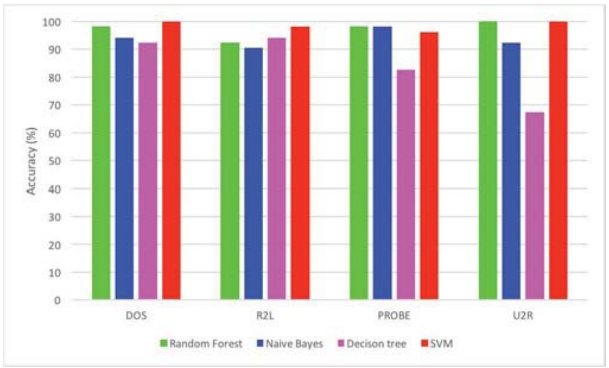

Figure: Accuracy vs Attack's category

Conclusions and Future Work

1. Random Forest is better than the other three algorithms in classifying attacks with a higher probability of detection, lower probability of false alarm, lower probability of miss detection, and higher accuracy
2. It detects most of the attack including DOS, PROBE, and U2R attacks. However, its main drawback is that it is time-consuming.
3. On the other hand, Naïve Bayes shows the lowest probability of detection, highest probability of false alarm, the highest probability of miss detection, and lowest accuracy; but, it requires less processing time than the three other algorithms.
4. As future work, we aim to develop algorithms that satisfy the tradeoff between processing time and accuracy.