Evaluation of Machine Learning Algorithms for Intrusion Detection System

By Srilalith Nampally

Introduction:

Intrusion detection systems are some of the most popularly used cyber-attack defense mechanisms on various cyber-physical and networked systems. I have a deep seated interest in security of computer systems and as such have already worked with ML based Intrusion Detection systems on SCADA systems, which is why I have selected this paper.

The study evaluates various machine learning classifiers on the KDD intrusion dataset focusing on false negative and false positive metrics to improve the intrusion detection system's performance.

The paper emphasizes the necessity of reliable intrusion detection systems (IDS) due to the increasing prevalence of harmful network attacks and the rapid development of information technology. It mentions various types of attacks like Denial of Service (DOS) and others, discussing the challenge in maintaining network integrity and confidentiality.

Origins of the KDD Dataset

The KDD Cup '99 dataset originates from the International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with the Fifth International Conference on Knowledge Discovery and Data Mining in 1999. The data was prepared by the Defense Advanced Research Projects Agency (DARPA) and managed by MIT's Lincoln Laboratory.

Structure of the Dataset

The KDD Cup '99 dataset was created based on data captured in a simulated military network environment that mimics a typical U.S. Air Force LAN. The network was instrumented with various types of attacks, executed during a controlled environment to generate a comprehensive dataset of network intrusions.

Key Features of the KDD Dataset:

- **Size and Content:** The dataset contains about 4.9 million records, each labeled as either normal or as an attack, with exactly one specific attack type.

- **Features:** Each record consists of 41 features. These features are derived from the data traffic attributes and can be classified into three groups:
    1. **Basic features:** Attributes that are derived from the packet headers without inspecting the payload, like duration, protocol type, and number of bytes.
    2. **Content features:** These features include assessments of the payload, such as the number of failed login attempts.
    3. **Traffic features:** These are computed with respect to a window interval and include the count of connections to the same host in the past two seconds.

Types of Attacks in the KDD Dataset

The attacks categorized in the dataset fall into four main categories:

1. **Denial of Service Attack (DoS):** The goal is to make a computer resource unavailable to its intended users, e.g., syn flood.
2. **Remote to Local Attack (R2L):** Unauthorized access from a remote machine, e.g., guessing password.
3. **User to Root Attack (U2R):** Unauthorized access to local superuser (root) privileges, e.g., various buffer overflow attacks.
4. **Probing Attack:** Surveillance and other probing, e.g., port scanning.

Technology & Methodology

Intrusion Detection System (IDS): IDS technology is designed to monitor network or system activities for malicious activities or policy violations. Any detected activity or violation is typically reported to an administrator or collected centrally using a security information and event management (SIEM) system. IDS technologies are primarily categorized into Network Intrusion Detection Systems (NIDS) and Host Intrusion Detection Systems (HIDS).

Machine Learning in IDS: The methodology of applying machine learning to IDS involves training algorithms to classify network behavior as either normal or malicious based on input data. This study utilizes the KDD Cup '99 dataset, a benchmark dataset for evaluating IDS efficiency. The dataset includes a wide variety of simulated attacks which creates an environment to test and train the IDS algorithms effectively.

The Machine Learning algorithms used in this paper:

J48 Decision Tree: The J48 classifier is an open-source implementation of the C4.5 algorithm in the Weka data mining tool. It creates a decision tree based on the concept of information entropy, iteratively selecting the best attribute that yields the most homogeneous sub-nodes.

Random Forest: A versatile and robust ensemble learning method, Random Forest constructs a multitude of decision trees at training time and outputs the mode of the classifications (or mean prediction) of the individual trees. It is less prone to overfitting than a single decision tree.

Decision Table: This classifier summarizes data into a "decision table" that lists combinations of attributes with corresponding classifications. It uses a majority rule approach for classification, providing a clear and concise summary of the decision logic.

Multilayer Perceptron (MLP): MLP is a class of feedforward artificial neural network (ANN). It consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Except for the input nodes, each node uses a nonlinear activation function. MLP utilizes a technique called backpropagation for training.

Naive Bayes: A simple yet effective classification technique based on Bayes' Theorem with an assumption of independence among predictors. It is particularly effective when dimensionality of the inputs is high, despite its assumption of an independent feature model.

Bayes Network: A Bayes Network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). It is used for building models that allow us to predict the likelihood of outcomes.

Experimental Setup:

In this paper, the experiments were performed on Ubuntu 13.10 platform, Intel R, Core(TM) i5-4210U CPU @ 1.70GHz (4CPUs), 6 GB RAM. Waikato Environment for Knowledge Analysis (WEKA) is a machine learning tool written in JAVA. It is an open source tool and available for free.

In order to implement a fair testing phase fully randomized 60000 have been extracted. The extracted testing data includes all 21 types of attacks within KDD dataset. There are several evaluations metrics can be used in a classification algorithm. In this paper, the confusion matrixes were generated for each machine learning classifiers.

Performance Metrics:

True Positive (TP): this value represents the correct classification attack packets as attacks.

True Negative (TN): this value represents the correct classification normal packets as normal.

False Negative (FN): this value illustrates that an incorrectly classification process occurs. Where the attack packet classified as normal packet, a large value of FN presents a serious problem for confidentiality and availability of network resources because the attackers succeed to pass through intrusion detection system.

False Positive (FP): this value represents incorrect classification decision where the normal packet classified as attack, the increasing of FP value increases the computation time but; on the other hand, it is considered as less than harmful of FN value increasing.

Precision: is one of the primary performance indica- tors. It presents the total number of records that are correctly classified as attack divided by a total number of records classified as attack. The precision can be calculated according to the following equation:

$$\boldsymbol{P = \frac{TP}{TP + FP}}$$

J48 tree classifier was tested with confidence factor = 0.25; numFolds = 3; seed = 1; unpruned = False, collapse tree = true and sub tree rising =true. Random forest classifier also tested with number of trees =100 and seed =1. Random tree classifier was tested with min variance = 0.001 and seed = 1. A decision table classifier was tested based on the Best First Search (BFS) and cross value = 1. Furthermore, the MLP classifier was tested with the following parameters: search learning rate=0.3, momentum =0.2, validation threshold=20.

Results:

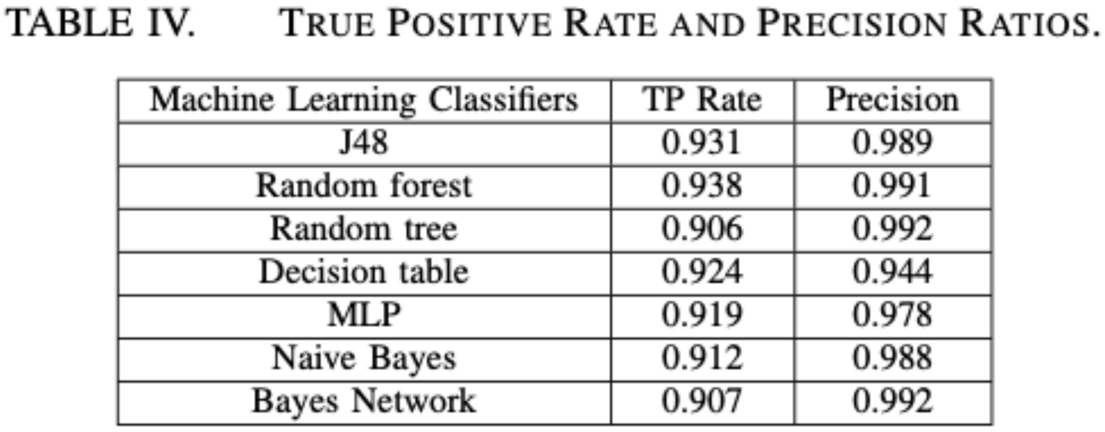
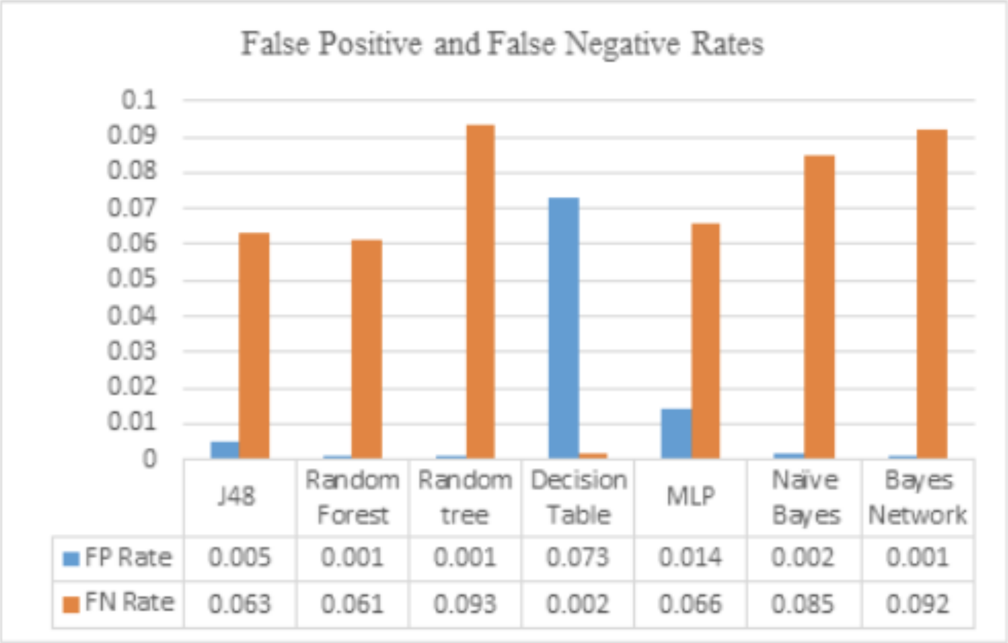


TABLE IV. TRUE POSITIVE RATE AND PRECISION RATIOS.

Machine Learning Classifiers	TP Rate	Precision
J48	0.931	0.989
Random forest	0.938	0.991
Random tree	0.906	0.992
Decision table	0.924	0.944
MLP	0.919	0.978
Naive Bayes	0.912	0.988
Bayes Network	0.907	0.992



	J48	Random Forest	Random tree	Decision Table	MLP	Naïve Bayes	Bayes Network
🟦 FP Rate	0.005	0.001	0.001	0.073	0.014	0.002	0.001
🟧 FN Rate	0.063	0.061	0.093	0.002	0.066	0.085	0.092

TABLE V. ROOT MEAN SQUARE AND AREA UNDER THE RECEIVER OPERATING CHARACTERISTIC.

Machine Learning Classifiers	ROC Area	Root Mean Squared Error
J48	0.969	0.0763
Random forest	0.996	0.0682
Random tree	0.953	0.0763
Decision table	0.984	0.0903
MLP	0.990	0.0813
Naive Bayes	0.969	0.0872
Bayes Network	0.997	0.0870

The total number of incorrectly classified records for each selected classifiers are presented in the Table VI. The average accuracy rate is calculated by the following formula:

$$Average\ Accuracy\ Rate = \frac{TP + TN}{TP + FN + FP + TN}$$

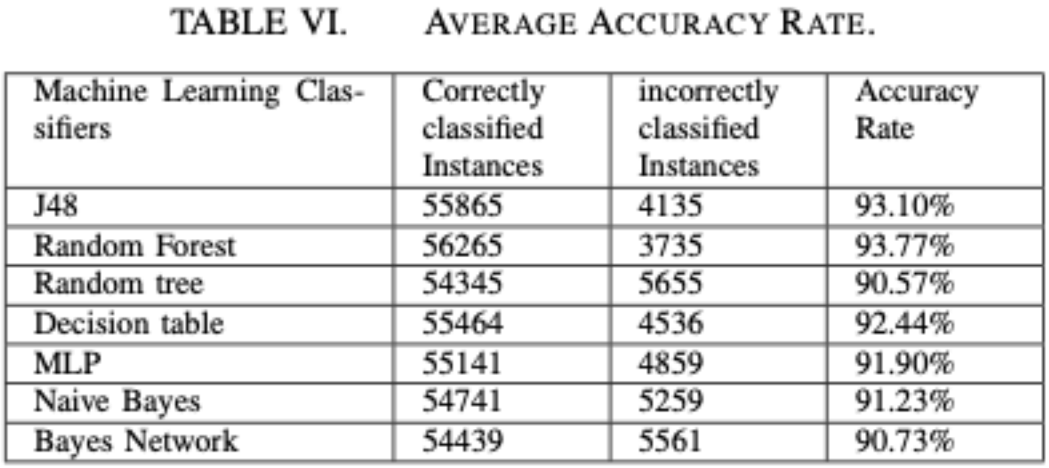


TABLE VI. AVERAGE ACCURACY RATE.

Machine Learning Classifiers	Correctly classified Instances	incorrectly classified Instances	Accuracy Rate
J48	55865	4135	93.10%
Random Forest	56265	3735	93.77%
Random tree	54345	5655	90.57%
Decision table	55464	4536	92.44%
MLP	55141	4859	91.90%
Naive Bayes	54741	5259	91.23%
Bayes Network	54439	5561	90.73%

Inference of Results:

- The Random forest achieved the highest accuracy rate 93.77 with smallest RMSE value and false positive rate.
- The Random tree classifier reached the lowest average accuracy rate 90.73 with smallest ROC value.
- Regarding to the average accuracy rate there is no big difference between MLP classifier and Naive Bayes classifier.
- All machine learning classifiers present acceptable precision rates for detecting normal packets.
- Bayes network classifier recorded the highest value for detecting correctly the normal packet.
- There are no big differences between MLP and J48 classifiers based on FN parameters.
- The decision table classifier did not reached the highest accuracy rate, but it had the lowest FN rate and it has a low time demand for building the training model.

Conclusion:

The study concludes that while the effectiveness of an IDS can be significantly enhanced by selecting appropriate machine learning algorithms, the choice of classifier depends heavily on the specific requirements and constraints of the network environment being protected. The paper advocates for a balanced approach to choosing an IDS classifier, considering both the potential for false alarms and the consequences of missed detections.

How this paper benefited me:

By going through this paper, I got a better insight into how Machine learning algorithms can be used for enhancing Intrusion detection systems. The data preprocessing steps were also innovative, specifically how they obtained the statistical data. With the new perspective and information I got from this paper, perhaps I can revisit my project on Integration of Statistical and ML based IDS and enhance its performance.

Critical Review:

The paper provides a comprehensive evaluation of several machine learning techniques for improving IDS effectiveness. However, it could further benefit from a deeper analysis of the computational efficiency of these classifiers and their scalability in real-world scenarios. Moreover, exploring the integration of hybrid models combining the strengths of multiple classifiers could be a valuable direction for future research.