**TITLE**

Anomaly-Based Network Intrusion Detection System through Feature Selection and Hybrid Machine Learning Technique

**INTRODUCTION**

An **intrusion detection system** (**IDS**) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any intrusion activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources and uses alarm filtering techniques to distinguish malicious activity from false alarms. Network intrusion detection systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet, and matches the traffic that is passed on the subnets to the library of known attacks. Once an attack is identified, or abnormal behaviour is sensed, the alert can be sent to the administrator. An example of a NIDS would be installing it on the subnet where firewalls are located in order to see if someone is trying to break into the firewall. Ideally one would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network.

In this paper, we focus on NID’s binary classification problem where the system differentiates between normal or attack activities. We propose an anomaly-based network intrusion detection system based on a combination of feature selection, K-Means clustering and XGBoost classification model. The reason behind selection of XGBoost classifier model is due to its strong performance, a variety of hyperparameter selection, fast implementation and popularity among machine learning communities. We test the performance of our proposed system over NSL-KDD dataset using KDDTest+ dataset. A feature selection method based on Attribute Ratio (AR) is applied to construct a reduced feature subset of NSL-KDD dataset. Moreover, due to feature selection, our proposed model employs only 75 out of 122 features (61.47%) to achieve this level of performance comparable to those using full number of features to train the model.

**DATASET DESCRIPTION**

Here we have used the NSL-KDD Train+ dataset for training purpose and for testing purpose we have used NSL-KDD Test+ dataset. NSL-KDD dataset contains 41 features categorizing into 3 types consisting of 3 nominal features, 6 binary features and 32 numeric features. The NSL-KDD categorizes attacks into 4 types consisting of Denial-of-Service, Probe, Root to Local and Unauthorized to Root as presented in Table 1. Each type of attack can be explained as follows:

1) Denial-of-Service (DoS): This type of attack overwhelms the targets’ resources (Network, CPU or Memory) so that typical operations cannot be performed as expected. Examples of this attack include sending huge number of packets to the targeted server so that normal users cannot access. 2) Probe: This type of attack involves port scanning to identify vulnerabilities in computer systems for further attacks.

3) Root to Local (R2L): The attackers try to access the unauthorized computer resources in order to destroy or modify operations of the targeted computer systems.

4) Unauthorized to Root (U2R): The attackers try to gain accesses to unauthorized resources using root privileges.

|  |  |
| --- | --- |
| Category | Attacks |
| DoS (Denial of Service) | Neptune, pod, smurf, teardrop, process table, warezmaster, apache2, mail bomb, back |
| Probe | multihop, http tunnel, ftp\_write, root kit, ps buffer overflow, xterm |
| R2L (Root to Local) | named, snmpgetattack,xlock, send mail, guess\_passwd |
| U2R (Unauthorized to Root) | ipsweep, nmap, port sweep, satan, mscan, saint |

Table 1. Type of Attacks

**DATASET PRE-PROCESSING**

1. **One Hot Encoding and Scaling**: We exercise One-Hot Encoding to transform 3 nominal features listed as protocol\_type, service and flag into 84 binary features (protocol has 3 features, service has 70 features and flag has 11 features). In summary, after One-Hot Encoding process, there are 122 features entering Normalization process. During Normalization process, we scale the dataset so that the mean of every feature is equal to zero and standard deviation is equal to one.
2. **Feature Selection**: To enhance model efficiency, reduce computational complexity and remove irrelevant features, we implement feature selection based on calculating the average AR. This value will be used to determine the feature importance of every feature and its calculation can be explained as follows. In AR approach, we employ attribute average and frequency for numeric and binary features, respectively. The AR of the jth feature AR (j) can be calculated as

AR(j)= max(i∈ [0,4]) CR(I,j) (1)

where CRi(j) is Class Ratio of the jth feature of the ith label( i∈ [0,4] where i = 0 for Normal, i=1 for DoS, i=2 for Probe, i=3 for R2L and i=4 for U2R class). For jth numeric feature, CRi(j) can be expressed as

CR(I,j)=AVG(I,j)/AVG(T,j) (2)

whereAVGi,j=Ci,j/Ni,j. Ci,j is the sum of the jth feature corresponding to ith label and Ni,j is the number of records of the jth features corresponding to the ith label.

AVG(T,j)= ∑i C(i,j)/n (3)

is the sum of jth feature divided by the total number of jth feature (Nj). For jth binary feature, CRi(j) can be written as

CR(I,j)=Freq(1)i,j/Freq(0)i,j (4)

where Freq (1) i,j is the number of ith records whose jth feature is equal to one and Freq(0)i,j is the number of ith records whose jth feature is equal to zero. Figure 2 displays top ten highest-important features obtained from Eq. (1). Features whose AR values less than 0.01 are removed from the analysis. The threshold 0.01 is judiciously selected to obtain the best performance with acceptable computational complexity. After applying feature selection method with threshold equal to 0.01, only 75 out of 122 features (61.47%) are left to be used to train the model.

**MODEL USED: K-MEANS CLUSTERING + XGBOOST CLASSIFIER**

1. **K- Means clustering**: The main objective of applying K-Means clustering to NSL-KDD dataset is to group a set of normal and attack traffic that exhibit similar pattern into the same partitions. Then, ML model corresponding to each partition is trained to differentiate normal or attack data within that group.
2. **XGBoostClassifier:** XGBoost was designed for speed and performance based on gradient-boosted decision trees algorithms. It provides the benefit of algorithm enhancement, model tuning, and can also be deployed in different computing environments. In addition, it allows the addition or tuning of regularization parameters to mitigate the impact of over-fitting.

**RESULTS**

After applying the hybrid model to train on NSL-KDD Train+ dataset we predicted the following output on the NSL-KDD Test+ dataset:

|  |  |  |  |
| --- | --- | --- | --- |
| **MODEL** | **WEIGHTED F1-SCORE** | **MACRO F1-SCORE** | **ACCURACY** |
| CLUSTERING+XGBOOST | 0.69 | 0.46 | 0.71 |
| CLUSTERING+DECISION TREE | 0.23 | 0.21 | 0.34 |
| CLUSTERING+SVM | 0.02 | 0.04 | 0.11 |
| XGBOOST WITHOUT CLUSTERING | 0.66 | 0.45 | 0.70 |