CMPLEXITY

STATISTICAL AND MACHINE LEARNING APPROACHES TO INTRUSION DETECTION SYSTEMS

SUPERVISED BY: DR. MAHA AMIN HASSANEIN

ABSTRACT

Traditional intrusion detection systems (IDS) struggle to detect novel cyber threats due to reliance on predefined signatures. This research develops an adaptive, statistical-based IDS that learns and models normal network behavior to identify anomalies in real-time. By leveraging advanced statistical techniques and machine learning algorithms like Random Forest, we aim to improve threat detection accuracy while minimizing false positives and negatives.

PROBLEM DEFINITION

The 2003 Slammer worm's rapid compromise of 90% of vulnerable systems in 10 minutes underscores the limitations of current rule-based intrusion detection systems (RBID). These systems often fail to detect new or subtle threats, struggle with high false positives, and lack real-time adaptability. The proposed solution addresses these issues by implementing a statistical and machine learning-driven IDS. It dynamically learns normal network traffic patterns, uses advanced statistical metrics to detect anomalies, employs ensemble learning for stronger detection, and offers near real-time threat identification.

DATASET DESCRIPTION

- **RRE-KDD Dataset**
  - Comprehensive network traffic collection
  - Includes 37 types of attacks (DoS, U2R, etc.)
  - Key variables: Protocol type, service, source/destination bytes...
  - Training set: 126,000 events
  - Testing set: 22,000 events

RANDOM FORESTS

Random Forest is an ensemble learning method for classification and regression. It builds multiple decision trees and combines their outputs for better accuracy and robustness.

Mathematical Representation

Given N decision trees $\{h_1(x), h_2(x), ..., h_N(x)\}$ the final result is

Classification: $\hat{y} = \text{mode}(h_i(x))$

Regression: $\hat{y} = \frac{1}{N}\sum_{i=1}^{N} h_i(x)$

Each tree is trained on a bootstrapped sample and splits nodes based on Gini impurity (G):

$$G = 1 - \sum_{i=1}^{C} p_i^2$$

where p_i is the probability of class i.

PRESENTED BY

- Amr Hany Sayed
- Tasneem Ahmed
- Abdelrahman Medhat
- Hussein Mohamed
- Youssef Mohamed
- Kareem Yasser

STATISTICAL APPROACHES

1 DESCRIPTIVE STATISTICS

Use mean and median to establish baseline network behavior

$$Mean = \frac{\sum_{i=1}^{n} x_i}{n}$$

Example: Tracking average packet size and login frequencies

2 THRESHOLD DETECTION

Standard Deviation Method

- Flagging criteria: $|x - Mean| > 3 \cdot StandardDeviation$

Standard Deviation:

$$(\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \text{Mean})^2}{n}})$$

Percentile Approach:

Unusual traffic: Data points in the top (100 - P%), where (P = 99%).

3 T-TEST ANALYSIS

$$t = \frac{x - \mu}{\frac{\sigma}{\sqrt{n}}}$$

x : Sample mean or individual data point.
μ : Population mean.
σ : Population standard deviation.
n : Sample size

4 CHI-SQUARE

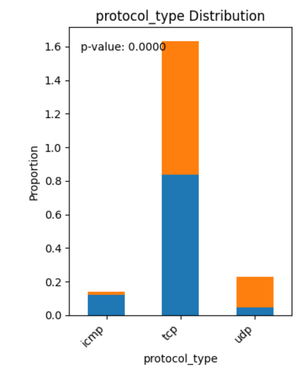
- Compare observed vs. expected network traffic distributions
- Determine statistical significance of traffic pattern deviations

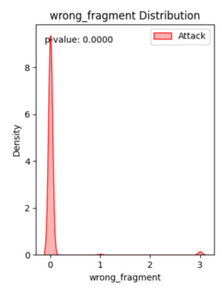
$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

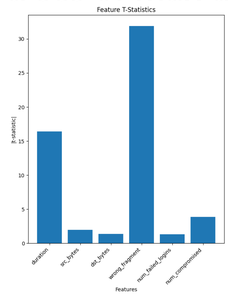
O_i: Observed frequency.
E_i: Expected frequency.
k : Number of categories.

EXPERIMENTAL WORK

- **Data preprocessing:**
  - Preprocessing steps were applied to the data to make it appropriate for the model and statistical analysis.
- **Statistical Tests** where employed to determine the top important features.
- **Random Forest**
  - **Encode Categorical Data**
    - Conversion of categorical variables to numeric values
  - **Preprocess the Data And Model Training**
    - Separating features and target variables
    - Converting target labels to binary (normal vs. attack)
    - Scaling numerical features using StandardScaler
    - Start Training the model with 100 trees of max depth = 10







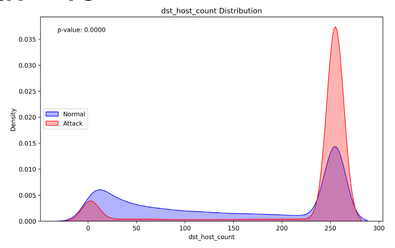


Figure 1: Density Distribution of a feature

RESULTS

- Based on T-test and Chi square
  - Most features show statistically significant differences between normal and attack traffic
  - Some features (DST_BYTES, NUM_FAILED_LOGINS, SRV_COUNT, NUM_COMPROMISED, ROOT_SHELL) have inconsistent significance between train and test datasets
  - Categorical features (PROTOCOL_TYPE, SERVICE, FLAG, LOGGED_IN, IS_GUEST_LOGIN) show strong statistical significance
  - Numerical features like DURATION, COUNT, and RERROR_RATE show substantial differences in means between normal and attack traffic
- Random forest Results

20% Training Data	precision	recall	f1-score	support
Normal	99%	99%	1	13422
Anomally	99%	98%	1	11773
		Accuracy: 99%		

KDDTest+	precision	recall	f1-score	support
Normal	66%	97%	0.78	9711
Anomally	97%	66%	0.75	12833
		Accuracy: 78%		

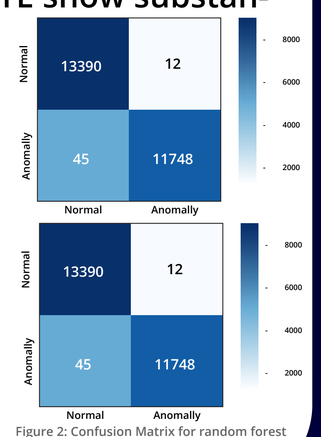


Figure 2: Confusion Matrix for random forest results on 20% of training data and on a seperate test dataset