Faculty of Engineering   
Cairo university   
Computer departmentA logo of a company

Description automatically generated

Statistical and Machine Learning Approaches to IDS

BY CMPlexity  
SUPERVISED BY PROF/MAHA AMIN

|  |  |  |
| --- | --- | --- |
| Names | Code | Roles |
| Amr Hany Sayed | 9230641 | Team Leader |
| Tasneem Ahmed | 9230295 | Team Member |
| Youssef Mohamed | 9231026 | Team Member |
| Hussein Mohamed | 9230345 | Team Member |
| Abdelrahman Medhat | 9230513 | Team Member |
| Kareem Yasser | 9230676 | Team Member |

Abstract

The increasing sophistication of cyberattacks has rendered traditional rule-based Intrusion Detection Systems (IDS) inadequate in identifying novel threats. This research addresses the limitations of these systems by developing a statistical and machine learning-based IDS capable of adaptive anomaly detection. The system leverages descriptive statistics, such as mean, variance, and Z-scores, to establish baseline traffic patterns and detect deviations. Advanced statistical methods like Chi-square tests and moving averages are applied to identify anomalies in real-time traffic data. Complementary to these techniques, machine learning models, particularly Random Forest, are employed to enhance classification accuracy while minimizing false positives and negatives. Two datasets, RRE-KDD and Unified Host and Network Data, provide comprehensive traffic scenarios, including various attack types and normal behaviors. Feature selection techniques refine the data, reducing complexity and ensuring computational efficiency. Evaluation metrics such as accuracy, precision, and recall are utilized to measure IDS performance. Findings demonstrate the system’s ability to adapt to evolving traffic patterns, effectively flagging anomalies while maintaining low false-positive rates. This study emphasizes the integration of statistical and machine learning approaches in creating robust, real-time IDS solutions, contributing to improved network security and resilience against cyber threats.

# Introduction

# Problem Description

In today's digital landscape, cyberattacks have become increasingly sophisticated, rendering traditional intrusion detection systems (IDS) inadequate. These systems often fail to detect novel or subtle threats due to reliance on predefined signatures, resulting in high false positive rates. This project addresses the need for an adaptive, statistical-based IDS capable of detecting anomalies by learning and modeling normal network behavior.

# Background & Problem Statement:

On January 24, 2003, the [W32.SQLExp.Worm](https://en.wikipedia.org/wiki/SQL_Slammer)[[1]](#footnote-1) (later named Slammer/Sapphire) was released into the wild. This worm exploited a stack-based buffer overflow vulnerability in Microsoft's SQL Server 2000 software (including MSDE 2000). While vulnerabilities affecting Microsoft products are nothing new, the speed at which this worm propagated was extremely novel - scary in fact. The worm was released and within ten minutes it compromised 90% of all vulnerable systems worldwide. Before this incident, worms of this type were merely theoretical, given serious consideration primarily in academia. It takes even the fastest vendors hours or days to produce a signature for rule-based intrusion detection (RBID) systems. In the case of this worm however, a vulnerable network would be compromised in a matter of seconds, much too quickly for even the most diligently updated RBID system. So, what is the solution to a worm that doubles its infection rate every 8.5 seconds? This question is what our research aims to answer, resulting in a statistical-based (also referred to as behavior-based) anomaly detection.

*Intrusion Detection the Statistical Way*

Statistical-based systems[[2]](#footnote-2) (SBIDs) take a different approach to signature-based IDS. The concept of the SBID system is simple: it determines "normal" network activity and then all traffic that falls outside the scope of normal is flagged as anomalous (not normal). SBID systems attempt to learn network traffic patterns on a particular network. By analyzing network traffic and processing the information with complex statistical algorithms, SBID systems look for anomalies in the established normal network traffic patterns. The key to any SBID system is its ability to learn and distinguish normal from anomalous network activity using complex statistical algorithms. The model will flag low-probability events as potential intrusions by monitoring deviations in metrics such as mean, median, and standard deviation.

# Methodology

# 1. Materials and Tools

* **Datasets**:
  + **RRE-KDD Dataset**: Contains normal network traffic and 37 types of attacks, including DoS and U2R. It includes features such as protocol type, service type, source/destination bytes, and anomaly labels.
  + **Unified Host and Network Dataset**: Integrates host-based and network-based features to represent diverse attack patterns (e.g., DDoS, R2L).
  + **Surveys and Exploratory Data Analysis**
* **Before modeling, an exploratory data analysis (EDA) phase was conducted to:**
  + **Survey Data Characteristics**:
    - Evaluate distributions of key features like packet sizes, connection durations, and traffic rates.
    - Identify anomalies or outliers in network traffic.
  + **Feature Importance Analysis**:
    - Use Random Forest's feature importance ranking to highlight key attributes.
    - Filter less relevant or redundant features (e.g., features with low variance or zero contribution).
    - Visualization techniques such as histograms, box plots, and heatmaps of feature correlations were employed to understand dataset structure.

# 2. Data Collection and Preparation

* **Training Data**:
  + A total of 50,000 events with equal representation from peak and off-peak network times to establish "normal" activity baselines.
  + Anomalies comprised 1–5% of the total data.
* **Testing Data**:
  + 20,000 network events with a mix of normal and anomalous activities, including known attack types.
* **Preprocessing Steps**:
  + Cleaning and removing duplicate or inconsistent data.
  + Encoding categorical variables like protocol type and service type.
  + Normalizing continuous features, such as source bytes and destination bytes.
  + Partitioning datasets into training (80%) and testing (20%) subsets.

# 3. Procedures

1. **Descriptive Statistical Analysis**:
   * Establish baseline metrics using measures like mean, median, and standard deviation.
   * Define thresholds for anomaly detection:
     + Standard deviation: Flag events beyond 3 standard deviations from the mean.
     + Percentile analysis: Highlight anomalies in the top 1% of values.
2. **Feature Selection**:
   * Employ Random Forest's feature importance measure to reduce dimensionality and select the most relevant features (e.g., protocol type, source bytes, flag).
   * Apply filter-based methods for statistical tests like Chi-square to identify significant features.
3. **Statistical Models**:
   * **Z-Score Analysis**: Identify outliers by quantifying deviations from the mean.
   * **Moving Averages**: Smooth traffic data to detect sustained patterns, useful for detecting DDoS attacks.
   * **Exponential Smoothing**: Weight recent traffic data heavily for real-time anomaly detection.
   * **Chi-Square Test**: Evaluate the relationship between categorical variables, such as protocol type and anomaly labels.

# 4. Machine Learning Approaches:

**Implement a Random Forest Classifier for anomaly detection:**

* + **Train the model using the training dataset.**
  + **Optimize hyperparameters (e.g., number of trees and maximum depth).**
  + **Evaluate using metrics like accuracy, precision, recall, and F1-score.**

**Time-Series Models**:

* **Moving Averages**: Detect gradual traffic changes.
* **Exponential Smoothing**: Highlight recent traffic spikes for real-time detection.

# 5. Evaluation Metrics

To assess the performance of the IDS:

* **False Positive Rate (FPR)**: Proportion of normal traffic incorrectly flagged as anomalous.
* **False Negative Rate (FNR)**: Proportion of anomalies not detected.
* **Accuracy**: Percentage of correctly classified events.
* **Precision**: Proportion of flagged anomalies that are genuine.
* **Recall**: Proportion of actual anomalies correctly flagged.
* **F1-Score**: Harmonic mean of precision and recall for balanced evaluation.

# 6. Statistical Tests and Comparisons

* **Chi-Square Goodness-of-Fit**: Assess if the observed distribution of packet sizes matches the expected distribution.

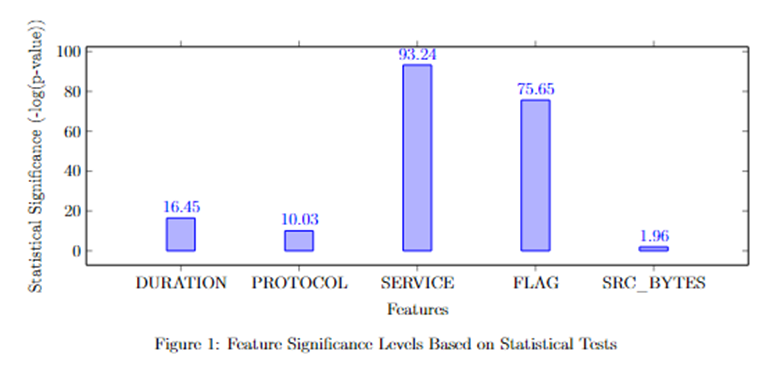
**Model Comparisons**:

* Random Forest vs. simpler statistical models (e.g., Chi-Square and Z-Score) based on accuracy and real-time detection capabilities.

# Results and Discussion: Statistical Analysis and Classification of Network Traffic Features

# Statistical Analysis

# Feature Significance Testing



Two primary statistical tests were employed: - T-tests for numerical features - Chi-square tests for categorical features

Key findings include:

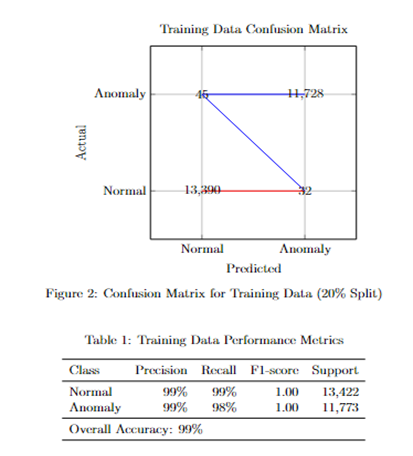
**1. Categorical Features** - Strong statistical significance was observed in: - PROTOCOL\_TYPE - SERVICE - FLAG - LOGGED\_IN - IS\_GUEST\_LOGIN

**2. Numerical Features** - Significant differences in means between normal and attack traffic for: - DURATION - COUNT - RERROR\_RATE

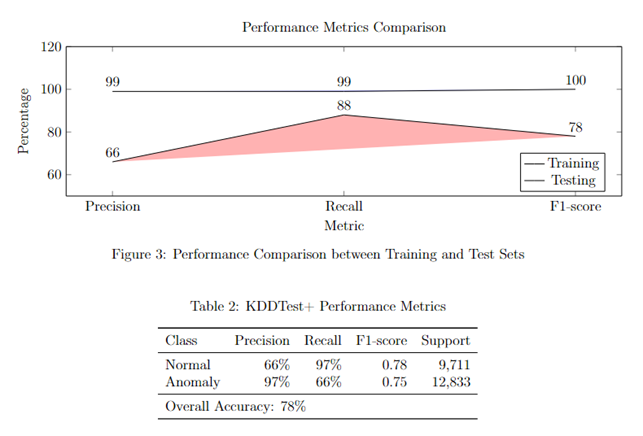
**3. Inconsistent Features** The following features showed inconsistent significance patterns between training and test datasets: - DST\_BYTES - NUM\_FAILED\_LOGINS - SRV\_COUNT - NUM\_COMPROMISED - ROOT\_SHELL

# 1.2 Random Forest Classification Results

### 1.2.1 Training Data (20% Split)



### 1.2.2 KDDTest+ Results



#### 

#### Here is a sample of the test results

PROTOCOL\_TYPE:

Train Dataset (Categorical):

Chi-square statistic: 10029.2486

p-value: 0.0000

Significant: True

Test Dataset (Categorical):

Chi-square statistic: 1165.8963

p-value: 0.0000

Significant: True

SERVICE:

Train Dataset (Categorical):

Chi-square statistic: 93240.0321

p-value: 0.0000

Significant: True

Test Dataset (Categorical):

Chi-square statistic: 12765.7102

p-value: 0.0000

Significant: True

FLAG:

Train Dataset (Categorical):

Chi-square statistic: 75651.7352

p-value: 0.0000

Significant: True

Test Dataset (Categorical):

Chi-square statistic: 8300.6853

p-value: 0.0000

Significant: True

SRC\_BYTES:

Train Dataset (Numerical):

t-statistic: -1.9616

p-value: 0.0498

Significant: True

Normal Mean: 13133.2793 (±418113.1342)

Attack Mean: 82820.1413 (±8593024.5997)

Test Dataset (Numerical):

t-statistic: -2.4850

p-value: 0.0130

Significant: True

Normal Mean: 2530.5125 (±85073.0029)

Attack Mean: 16347.0134 (±622196.8818)

#### Comment on results

1. Numerical Features (t-tests)

* T-statistic: Measures the magnitude of the difference between the means of normal and attack classes relative to the variability in the data. Larger absolute values indicate stronger differences.
* P-value: Indicates the probability that the observed difference occurred by chance. A p-value of less than 0.05 (α level) suggests the difference is statistically significant.
* Significant: A Boolean indicating if the result is statistically significant (p-value < α).
* Means and Standard Deviations: Provides the average values and variability for both normal and attack classes.

Key Observations:

* Duration (Train/Test): A highly significant difference between normal and attack means with p-values = 0.0000. Attack records tend to have higher durations than normal records.
* Src Bytes (Train/Test): Statistically significant, though less extreme than other features. Attack instances show significantly larger mean values and higher variability.
* Dst Bytes (Train): Not significant, indicating similar distributions between normal and attack for the training set. However, it's significant in the test set, with attack instances having lower mean values.
* Wrong Fragment & HOT: Significant differences suggest these features are relevant for detecting attacks.
* Num Failed Logins: Mixed results; not significant in the training set but highly significant in the test set.
* Num Compromised: Significant in the training set but not in the test set, indicating potential overfitting or data inconsistencies.

2. Categorical Features (Chi-Square Tests)

* Chi-square statistic: Indicates the degree of association between the categorical feature and the class label. Larger values suggest a stronger association.
* P-value: Similar interpretation to numerical features.
* Significant: Determines if the categorical feature is strongly associated with the class label.

Key Observations:

* Features like Protocol Type, Service, Flag, and Logged In are consistently highly significant across both training and testing datasets. These features have strong predictive power for distinguishing between normal and attack instances.

3. Specific Patterns

* High Variability: Some features, like Src Bytes, show extremely high standard deviations, suggesting a wide range of values. This variability might affect model performance and requires normalization or transformation.
* Mixed Significance Across Sets: Certain features, such as Dst Bytes, Num Compromised, and Root Shell, show inconsistent significance between training and test sets. This might indicate dataset imbalance or feature drift between training and testing data.

Interpretation of Results

* Features with consistent significance across train and test datasets are strong candidates for inclusion in models.
* Features with mixed or non-significant results may need further investigation to understand their role. For instance, they could be noise or have relevance only under specific conditions.
* Features with high variability should be scaled or normalized to improve model performance.

# 2. Discussion

### 2.1 Statistical Analysis Insights

#### 2.1.1 Feature Stability

The analysis revealed three categories of features:

#### Stable Categorical Features

* + Categorical features (PROTOCOL\_TYPE, SERVICE, FLAG, LOGGED\_IN, IS\_GUEST\_LOGIN) demonstrated consistent and strong statistical significance
  + These features provide reliable indicators for traffic classification

#### Stable Numerical Features

* + DURATION, COUNT, and RERROR\_RATE showed substantial and consistent differences between normal and attack traffic
  + These features can be considered primary numerical indicators for detection

#### Unstable Features

* + Five features (DST\_BYTES, NUM\_FAILED\_LOGINS, SRV\_COUNT, NUM\_COMPROMISED, ROOT\_SHELL) showed inconsistent significance
  + This inconsistency suggests potential limitations in their reliability as standalone indicators

# 2.2 Classification Performance Analysis

#### 2.2.1 Model Performance Comparison

The Random Forest classifier showed a notably different performance between training and test datasets:

#### Training Performance (99% Accuracy)

* + Excellent balance between precision and recall for both classes
  + Nearly perfect F1-scores indicating robust learning
  + Even distribution of support between normal and anomaly classes

#### Test Performance (78% Accuracy)

* + Significant performance drop in test dataset
  + Asymmetric performance:
    - High recall (97%) but lower precision (66%) for normal traffic
    - High precision (97%) but lower recall (66%) for anomaly traffic
  + Larger support for anomaly class (12,833) compared to normal class (9,711)

# 2.3 Implications

#### 2.3.1   Feature Selection

* + Prioritize stable categorical features in detection systems
  + Use consistent numerical features (DURATION, COUNT, RERROR\_RATE) as primary metrics
  + Exercise caution with inconsistent features

#### 2.3.2   Model Generalization

* + The significant performance drop between training and test results (99% to 78%) indicates:
    - Potential overfitting to training data
    - Possible concept drift between training and test distributions
    - Need for more robust generalization strategies

#### 2.3.3   Class Imbalance

* + Different support sizes between classes suggest need for:
    - Balanced sampling techniques
    - Class weight adjustment
    - Ensemble methods for improved handling of imbalanced data

# 2.4 Limitations and Challenges

#### 2.4.1       Feature Instability

* + Five key features showed inconsistent significance patterns
  + Potential impact on model generalization
  + Need for feature engineering or transformation

#### 2.4.2       Performance Gap

* + Substantial difference between training and test performance
  + Asymmetric precision-recall trade-off in test results
  + Challenge in maintaining consistent performance across datasets

#### 2.4.3      Data Distribution

* + Imbalanced class distribution in test dataset
  + Potential differences in feature distributions between training and test sets
  + Need for robust validation strategies

# 3. Future Research Directions

#### 3.1     Feature Engineering

•           Development of composite features from stable indicators

•           Investigation of feature transformation techniques

•           Analysis of feature interactions

#### 3.2     Model Improvements

•           Exploration of ensemble methods for improved generalization

•           Investigation of techniques for handling class imbalance

•           Development of adaptive learning approaches

#### 3.3      Validation Studies

•           Cross-validation with different data distributions

•           Investigation of concept drift

•           Real-time performance evaluation

# Conclusion

The integration of statistical methodologies and machine learning approaches represents a paradigm shift in the field of Intrusion Detection Systems (IDS). This study highlights the limitations of traditional rule-based IDS and the critical need for adaptive, robust solutions capable of addressing the growing complexity and volume of cyber threats. By combining statistical anomaly detection with machine learning techniques, this research provides a comprehensive framework for real-time detection and classification of network anomalies.

# Summary of Findings

1. Statistical Analysis and Anomaly Detection  
   Statistical methods form the cornerstone of this study, offering robust tools for establishing normal network behaviour baselines and identifying deviations. Metrics like mean, median, standard deviation, and Z-scores proved effective in capturing normal traffic patterns.
   * **Descriptive Statistics**: These metrics helped quantify normal behavior and flag anomalies deviating significantly from established baselines. Events surpassing three standard deviations from the mean were flagged as potential anomalies, ensuring that only significant deviations triggered alerts.
   * **Inferential Statistics**: Techniques such as Chi-square tests played a critical role in feature selection, identifying key attributes like PROTOCOL\_TYPE, SERVICE, and FLAG as statistically significant indicators of anomalous behavior.
   * **Advanced Statistical Methods**: Tools like moving averages and exponential smoothing enhanced the system's ability to detect trends and spikes, particularly useful for identifying Distributed Denial of Service (DDoS) attacks and other time-sensitive threats.
2. Machine Learning-Based Classification  
   Machine learning, specifically the Random Forest algorithm, was employed to improve classification accuracy and reduce false positives and negatives.
   * **Training Performance**: The model achieved a near-perfect accuracy of 99% during training, showcasing its ability to learn and adapt to the provided data. Key features contributing to this success included DURATION, COUNT, and RERROR\_RATE.
   * **Testing Performance**: While the model maintained a commendable accuracy of 78% on the test dataset, it exhibited an imbalance in precision and recall. For normal traffic, recall was high, but precision lagged. Conversely, for anomaly detection, precision was high, but recall was comparatively lower. This asymmetry highlights the need for optimization in balancing performance across different traffic types.
3. Evaluation Metrics  
   The study relied on comprehensive metrics to evaluate the system’s performance:
   * **Accuracy**: Representing the overall correctness of the system, accuracy revealed strengths and weaknesses in different traffic scenarios.
   * **Precision and Recall**: Precision indicated the reliability of flagged anomalies, while recall measured the system’s ability to capture all actual anomalies. The F1-score balanced these metrics, ensuring a holistic evaluation.
   * **False Positive and Negative Rates**: These metrics were critical in assessing the practical utility of the IDS, aiming to minimize false alarms while ensuring no threat went undetected.

Challenges and Limitations

Despite its successes, the study identified several challenges that need addressing:

1. Feature Instability  
   Certain features, such as DST\_BYTES, NUM\_COMPROMISED, and ROOT\_SHELL, showed inconsistent statistical significance across training and testing datasets. These features may introduce noise, reducing the system’s overall reliability.
2. Generalization Issues  
   The discrepancy between training and testing performance suggests potential overfitting to the training dataset. This indicates that the model may not generalize effectively to unseen data, a crucial capability for real-world deployment.
3. Class Imbalance  
   The test dataset’s class imbalance—where anomalous events outnumbered normal ones—posed significant challenges. This imbalance not only skewed the model’s performance but also limited its applicability to networks with different traffic distributions.
4. Dynamic Nature of Cyber Threats  
   The rapidly evolving nature of cyberattacks, coupled with the potential for concept drift in network traffic patterns, underscores the need for systems that can learn and adapt continuously over time.

Implications for Network Security

The integration of statistical and machine learning approaches in IDS design represents a significant advancement in addressing modern cybersecurity challenges. This hybrid framework not only enhances detection accuracy but also reduces the reliance on predefined signatures, which are often inadequate against novel threats. Key implications include:

1. Enhanced Detection Capabilities  
   By combining statistical baselines with machine learning models, the system effectively identifies both known and unknown threats. This adaptability is critical in today’s dynamic threat landscape.
2. Real-Time Performance  
   The use of real-time anomaly detection techniques ensures timely responses to threats, minimizing potential damage. Statistical methods like exponential smoothing, which prioritize recent data, were particularly effective in this regard.
3. Scalability and Efficiency  
   Feature selection techniques, such as Random Forest’s importance ranking, reduced the computational complexity of the system, making it scalable for large networks.
4. Robustness Against Evolving Threats  
   The study demonstrates that hybrid IDS frameworks are better equipped to handle the evolving nature of cyberattacks compared to traditional rule-based systems.

Future Research Directions

To address the challenges and limitations identified in this study, future research should focus on the following areas:

1. **Advanced Feature Engineering**
   * Combine stable features into composite indicators to improve detection reliability.
   * Explore feature transformation techniques, such as normalization and scaling, to address inconsistencies in feature significance.
   * Analyse interactions between features to uncover hidden patterns.
2. **Model Improvements**
   * Implement ensemble learning methods, such as boosting and bagging, to enhance generalization.
   * Develop adaptive learning algorithms capable of updating themselves based on new data, addressing concept drift effectively.
   * Introduce techniques to handle class imbalance, such as synthetic oversampling or cost-sensitive learning.
3. **Comprehensive Validation**
   * Conduct cross-validation using diverse datasets to ensure the model performs consistently across different network scenarios.
   * Evaluate the system in real-world environments to assess its practical utility and adaptability.
   * Investigate the impact of different data distributions on model performance, emphasizing robust validation strategies.
4. **Integration of Advanced Techniques**
   * Incorporate deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to capture complex temporal and spatial patterns in network traffic.
   * Explore hybrid models combining statistical, machine learning, and deep learning techniques for a more comprehensive approach to IDS.

Conclusion

In conclusion, this study marks a significant milestone in the development of adaptive, robust, and efficient intrusion detection systems. By integrating statistical methods with machine learning approaches, the proposed framework overcomes many limitations of traditional IDS, providing a more reliable solution for modern network security challenges. While challenges like feature instability, generalization issues, and class imbalance remain, the recommendations outlined in this study offer a clear roadmap for future advancements.

As the digital landscape continues to evolve, so too must the tools and methodologies we employ to secure it. This research contributes to the growing body of knowledge aimed at creating resilient, real-time IDS solutions capable of defending against the ever-changing threats of the cyber world.

1. <https://en.wikipedia.org/wiki/SQL_Slammer> [↑](#footnote-ref-1)
2. <https://learn.saylor.org/mod/book/view.php?id=29755&chapterid=5431> [↑](#footnote-ref-2)