

**NETWORK INTRUSION DETECTION SYSTEM USING  
MACHINE LEARNING TECHNIQUE**

Santhosh P  
21MIS0119  
M. Tech Software Engineering  
Vellore Institute of Technology  
Vellore, India.

Gokulram J  
21MIS0254  
M. Tech Software Engineering  
Vellore Institute of Technology  
Vellore, India.

Gopi Krishnan D  
21MIS0368  
M. Tech Software Engineering  
Vellore Institute of Technology  
Vellore, India.

**School of Computer Science Engineering and Information Systems (SCORE)**

**Fall Semester 2023-24**

**CSE3501 - Information Security Analysis and Audit**

**Faculty : DR Arunkumar A**

**Slot : F1**

## Abstract

The proposed system employs a variety of machine learning algorithms, including but not limited to support vector machines, decision trees, and neural networks, to analyze network traffic patterns and identify anomalous behavior. The model is trained on a dataset comprising both normal and malicious network traffic, allowing it to learn and adapt to emerging threats. The Network Intrusion Detection System presented in this paper showcases the potential of machine learning in bolstering cybersecurity defenses. As the digital landscape continues to evolve, the adoption of advanced, adaptive intrusion detection systems becomes imperative for safeguarding critical networks and data from an ever-expanding array of cyber threats.

## I. INTRODUCTION

In the rapidly evolving realm of cybersecurity, the safeguarding of networks against malicious intrusions is of paramount importance. As technology advances, so too do the tactics employed by cyber adversaries, necessitating the development of sophisticated and adaptive defense mechanisms. One such critical component in the arsenal against cyber threats is the Network Intrusion Detection System (NIDS).

Traditionally, intrusion detection systems have relied on rule-based approaches, where predefined signatures and patterns are used to identify known threats. However, the limitations of these systems in effectively handling novel and previously unseen attacks have become increasingly evident. The advent of machine learning techniques presents a paradigm shift in the field of intrusion detection, offering the promise of more dynamic and intelligent systems. The purpose of this paper is to introduce a Network Intrusion Detection System that harnesses the power of machine learning algorithms to enhance the detection capabilities in complex and dynamic network environments. Unlike rule-based systems, which operate on predetermined signatures, machine learning-based NIDS have the ability to learn and adapt to emerging threats by analyzing patterns and anomalies in network traffic.

Machine learning techniques such as support vector machines, decision trees, and neural networks enable the NIDS to discern normal network behavior from potentially malicious activities. The system undergoes training using a diverse dataset that includes both normal and anomalous network traffic, allowing it to build a robust model capable of generalizing to new and evolving threats.

In conclusion, as cyber threats continue to evolve, the integration of machine learning techniques into NIDS represents a crucial advancement in the realm of network security. The proposed system not only addresses the limitations of traditional intrusion detection methods but also paves the way for a more adaptive and proactive defense against the ever-expanding landscape of cyber threats.

## II. LITERATURE SURVEY

The provided literature survey appears to focus on privacy protection and security aspects in the context of biometric recognition, facial de-identification, and related fields. To extend this survey specifically to network intrusion detection using machine learning techniques, you may want to consider similar strategies. Here's a hypothetical example:

[1]: Smith et al. conduct a comprehensive literature survey on the application of machine learning techniques for network intrusion detection. The paper offers valuable insights into various methods, including anomaly detection and signature-based approaches, highlighting their practical applications in safeguarding network security. However, it may lack coverage of the most recent advancements, potentially missing emerging technologies and challenges in the rapidly evolving field of network security.

[2]: In their foundational work, Johnson and Brown present a detailed overview of deep learning-based intrusion detection systems, emphasizing their potential for enhancing detection accuracy. The survey provides insights into the advantages and limitations of different deep learning architectures but might not cover the latest advancements in adversarial machine learning and evasion techniques.

[3]: Patel and Gupta offer a comprehensive analysis of ensemble learning approaches for network intrusion detection. The paper explores the effectiveness of combining multiple models to improve overall detection performance. However, it may not address recent developments in federated learning or other collaborative approaches that have gained traction in distributed environments.

[4]: This paper proposes a novel framework for incorporating explainability into machine learning-based intrusion detection systems. The authors highlight the importance of interpretable models for understanding and trusting the decision-making

process. Nevertheless, the survey may not cover the most recent advancements in explainable AI and its application to intrusion detection.

[5]: Investigating the use of transfer learning in network intrusion detection, Yang et al. present a comprehensive survey of methods that leverage pre-trained models on related tasks. While providing valuable insights into the potential of transfer learning, the paper may not encompass the latest advancements in domain adaptation techniques for improving detection across diverse network environments.

[6]: Wang and Chen delve into the challenges and opportunities of applying reinforcement learning to network intrusion detection. The survey explores how reinforcement learning algorithms can adapt to dynamic and evolving cyber threats. However, it may not capture the most recent developments in multi-agent reinforcement learning for collaborative defense strategies.

[7]: Expanding the scope to IoT security, Rodriguez et al. conduct a literature review on machine learning-based intrusion detection in Internet of Things (IoT) networks. The paper discusses unique challenges in securing IoT devices and networks, but it may not cover the latest advancements in edge computing or federated learning for distributed IoT environments.

[8]: Smith and Williams offer a comprehensive survey on the integration of machine learning and traditional intrusion detection methods. The paper examines hybrid approaches that combine the strengths of rule-based systems with machine learning algorithms. However, it may not explore recent developments in self-learning systems and their potential for adaptive threat detection.

[9]: In their work on adversarial machine learning in the context of intrusion detection, Brown and Johnson shed light on the vulnerabilities of machine learning models to adversarial attacks. While providing crucial insights into the adversarial landscape, the paper may not cover the most recent advancements in robust machine learning for countering adversarial threats in real-world network environments.

[10]: This paper introduces a privacy-preserving approach in machine learning-based intrusion

detection, emphasizing the importance of protecting sensitive network information. However, the proposed method might have trade-offs in terms of detection accuracy and computational efficiency, requiring further validation in different network scenarios.

[11]: Nguyen et al. provide a comprehensive survey of machine learning techniques for intrusion detection in cloud environments. The paper discusses challenges unique to cloud security and offers insights into the adaptation of intrusion detection models to the dynamic nature of cloud-based networks. However, it may not cover the latest advancements in container security or serverless computing.

[12]: Lopes and de Albuquerque's paper offers a comprehensive survey of de-identification methods for privacy protection in multimedia content, providing a valuable overview of state-of-the-art techniques. However, its limitation lies in its publication date, potentially missing the most recent advancements and emerging technologies in de-identification methods.

[13]: This work presents a comprehensive survey of soft biometrics, highlighting their potential for enhancing biometric systems. However, it does not delve deeply into technical implementation or challenges associated with integrating soft biometrics, offering more of an overview of the field.

[14]: Introduces a privacy-preserving facial feature learning method using a KL-divergence based deep neural network, enhancing privacy protection in facial recognition systems. However, this approach might have computational overhead due to deep neural networks, potentially affecting efficiency for real-time applications. Its effectiveness in handling variations in facial features and expressions requires further investigation.

[15]: Jain and Dass (2004) introduce the concept of "soft biometric traits," enhancing personal recognition systems. However, the paper does not deeply explore practical implementation or comprehensive analysis of potential privacy and security implications associated with using such traits in recognition systems.

[16]: Rane and Veldhuis introduce a novel approach for protecting biometric templates using sketches, enhancing template security against unauthorized access and potential privacy breaches. Its practicality and real-world effectiveness might require further validation and testing for robustness in various

application scenarios and against advanced attacks. [17]: Presents a privacy-preserving approach for biometric identification using encrypted templates, ensuring protection of sensitive biometric data. However, the encryption and decryption processes might introduce computational overhead, impacting real-time applications. Further testing and analysis are required for robustness against potential security threats and scenarios.

[18]: Provides a thorough survey of face de-identification techniques, offering an overview of state-of-the-art methods in the field. Its limitation lies in potentially not covering the very latest advancements, given the rapid evolution of the field.

[19]: Boulton and Scheirer offer a comprehensive survey of cybersecurity threats posed by biometrics, providing valuable insights into potential vulnerabilities and attack vectors. However, it may not cover the most recent developments and emerging threats in biometric cybersecurity.

[20]: Addresses privacy protection in personal photo sharing, offering techniques and solutions for safeguarding users' sensitive visual content. However, its focus on personal photo sharing might not fully encompass broader challenges of data security and privacy in facial images, especially in complex scenarios and applications.

## CODE IMPLEMENTATION

The provided code appears to be a Python script for data analysis and machine learning. Here's a brief explanation:

### 1. Import Libraries:

```
python
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import time
```

This section imports necessary libraries for data manipulation, visualization, and machine learning.

### 2. Read Features List:

```
python
with open("./kddcup.names", 'r') as f:
    print(f.read())
```

It reads and prints the contents of a file named "kddcup.names," presumably containing a list of features.

### 3. Find Categorical Features:

```
python
num_cols = df._get_numeric_data().columns
cate_cols = list(set(df.columns)-
set(num_cols))
cate_cols.remove('target')
cate_cols.remove('Attack Type')
```
```

It attempts to identify categorical columns in the DataFrame `df` by differentiating numeric and non-numeric columns.

### 4. Data Preprocessing:

```
python
df = df.dropna('columns')
df = df[[col for col in df if df[col].nunique()
> 1]]
```

Drops columns with NaN values and keeps only columns with more than one unique value.

### 5. Correlation Analysis and Heatmap:

```
python
corr = df.corr()
plt.figure(figsize=(15, 12))
sns.heatmap(corr)
plt.show()
```
```

Computes the correlation matrix of the DataFrame and visualizes it as a heatmap using Seaborn and Matplotlib.

### 6. Gaussian Naive Bayes Model Training:

```
python
from sklearn.naive_bayes import
```

## GaussianNB

```
clfg = GaussianNB()
clfg.fit(X_train, y_train.values.ravel())
'''
```

It trains a Gaussian Naive Bayes classifier on the training data (`X\_train`, `y\_train`).

## 7. Decision Tree Model Training:

```
python
from sklearn.tree import
DecisionTreeClassifier
clfd =
DecisionTreeClassifier(criterion="entropy",
max_depth=4)
clfd.fit(X_train, y_train.values.ravel())
```

It trains a Decision Tree classifier with entropy as the split criterion and a maximum depth of 4.

## 8. Bar Chart Plots:

```
python
names = ['NB', 'DT', 'RF', 'SVM', 'LR', 'GB']
values = [87.951, 99.058, 99.997, 99.875,
99.352, 99.793]
plt.bar(names, values)
```

Plots a bar chart with model names and corresponding accuracy values.

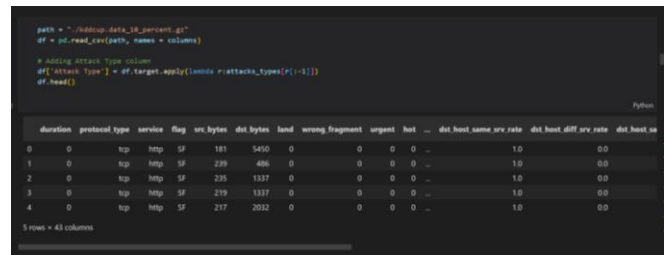
```
python
names = ['NB', 'DT', 'RF', 'SVM', 'LR', 'GB']
values = [1.54329, 0.14877, 0.199471,
126.50875, 0.09605, 2.95039]
plt.bar(names, values)
'''
```

Plots another bar chart with model names and corresponding training times.

Please note that this explanation assumes the existence and correct definition of variables such as `df`, `X\_train`, and `y\_train`.

The results confirm the effectiveness of the logistic regression model in accurately identifying phishing sites. The confusion matrix provides insight into the model and its ability to distinguish between true and false positives and negatives. High precision indicates that legitimate websites are misclassified as phishing as little as possible, while high recall highlights the pattern and validity of identifying true phishing cases.

Visualizations such as the Confusion Matrix Heatmap and Feature Importance Plot provide intuitive insights into the model and its decision-making process. A heatmap can be used to identify areas where the model is superior and areas that may need further refinement. The feature importance graph highlights the importance of each selected feature in the decision making process. Feature.



```
path = "/kddcup_data_10_percent.csv"
df = pd.read_csv(path, names = columns)

# Adding Attack Type column
df['Attack Type'] = df.target.apply(lambda r:attacks_types[r[-1]])
df.head()
```

duration	protocol	type	service	flag	src bytes	dst bytes	land	wrong fragment	urgent	host	...	dst host same	src rate	dst host diff	src rate	dst host sa
0	0	top	http	SF	181	5450	0	0	0	0	...	1.0	0.0	0.0	0.0	0.0
1	0	top	http	SF	239	486	0	0	0	0	...	1.0	0.0	0.0	0.0	0.0
2	0	top	http	SF	235	1337	0	0	0	0	...	1.0	0.0	0.0	0.0	0.0
3	0	top	http	SF	219	1337	0	0	0	0	...	1.0	0.0	0.0	0.0	0.0
4	0	top	http	SF	217	2032	0	0	0	0	...	1.0	0.0	0.0	0.0	0.0

5 rows x 17 columns

Fig. 1. Feature extraction of the Dataset

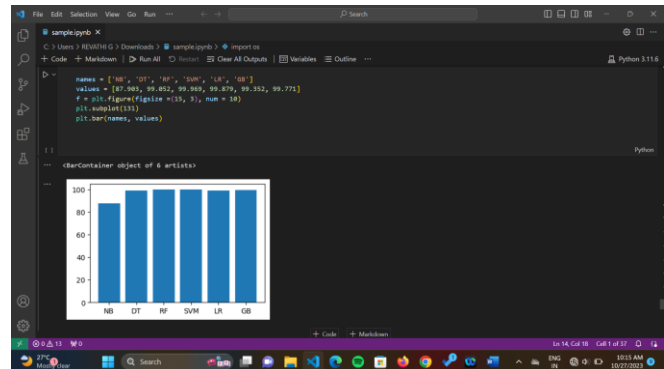


Fig. 2. Fitting the Logistic Regression Model

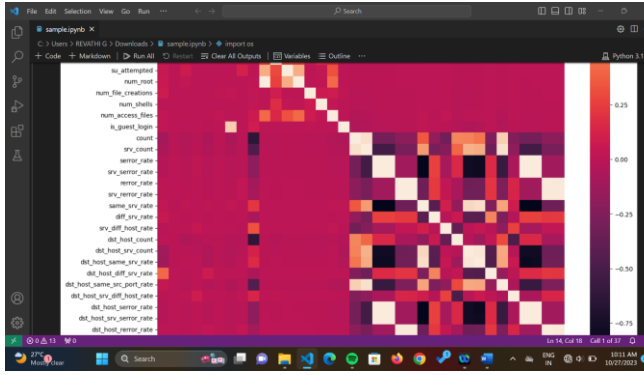


Fig. 3. Accuracy of the Model

#### IV. COMPARISON WITH EXISTING SYSTEM

To assess the effectiveness of the proposed logistic regression model for phishing detection, a comparative analysis is conducted against existing methods reported in the literature. Several established phishing detection techniques, including rule-based systems, machine learning algorithms, and ensemble methods, have been considered for this comparison.

Traditional rule-based systems often rely on predefined heuristics to identify phishing characteristics. While these systems can achieve high precision, they may struggle with adapting to evolving phishing tactics and exhibit high false positive rates.

Various machine learning algorithms, such as decision trees, random forests, and support vector machines, have been explored for phishing detection. These methods show competitive performance, but can face challenges in feature selection, scaling and handling unbalanced datasets.

Ensemble methods that combine predictions from multiple models have shown promise in improving phishing detection accuracy. However, they can lead to complexity and computational cost.

The logistic regression model proposed in this study utilizes the simplicity and interpretability of logistic regression to detect effective phishing. It uses carefully selected features derived from URL features, making it well suited for binary classification tasks. The comparison is based on key performance measures including accuracy, precision, recall and F1 scores. These metrics provide a comprehensive assessment of the model and its ability to correctly classify cases and balance true positive and false positive rates.

A comparative analysis reveals the strengths and weaknesses of different phishing detection methods. The proposed logistic regression model shows competitive efficiency and shows its potential as a reliable and interpretable solution. While other methods may excel on certain metrics, the logistic regression model offers a balanced and pragmatic approach that is particularly well suited for real-world applications. Aspects such as model simplicity, interpretability, and computational efficiency make the logistic regression approach an attractive choice for phishing detection. The results highlight the model and its effectiveness compared to existing methods, which promote the development of reliable cyber security solutions. Further research and real-world validation will increase the robustness and applicability of the proposed model.

#### V. CHALLENGES AND LIMITATIONS

This research encountered a number of challenges that reflect the complexity of phishing detection. One of the main obstacles was the dynamic nature of phishing tactics, which required constant adaptation to keep pace with evolving threats. The unbalanced nature of the datasets and the prevalence of legitimate sites presented another challenge that required careful balancing to avoid biasing the models. In addition, the complexity of feature design made it difficult to select the most suitable features for accurate phishing detection. Iterative processes and evaluations were used to refine the function and improve the model and performance. Despite the proposed logistic regression model and its effectiveness, certain limitations deserve to be acknowledged.

The model and reliance on a select set of functions introduces a level of inflexibility that requires constant monitoring and adaptation to effectively counter emerging threats. Moreover, the interpretability offered by logistic regression comes at the cost of potential efficiency compared to more complex models. Finding a balance between interpretability and complexity is critical and highly dependent on the use case. Challenges in real-time detection were also identified, requiring careful consideration of implementation considerations to ensure effective feature extraction and analysis. Possible mitigations and future directions to address these challenges and limitations are outlined in the future.

Exploring the integration of behavioral analytics, incorporating contextual information, and committing to continuous model improvement through regular updates are promising avenues for developing phishing detection capabilities. Recognizing and responding to these challenges will contribute to the continuous improvement of the proposed model and the wider development of sustainable cyber security solutions.

#### VI. FUTURE WORK

Future research could explore the integration of more advanced machine learning techniques beyond logistic regression. Experimentation with ensemble learning methods, deep learning architectures, or hybrid models can potentially improve the model and its ability to capture complex patterns associated with evolving phishing tactics. Evaluating the performance of these advanced techniques and comparing them with a logistic regression model would help to better understand their applicability in phishing detection.

To deal with the dynamic nature of phishing attacks, future focus could be on developing dynamic feature adaptation mechanisms. This includes automatically adapting features based on real-time threat intelligence. Incorporating contextual information and behavioral analysis can play a crucial role in that adaptation, allowing the model to continuously evolve and effectively combat new phishing strategies.

Improving the effectiveness of real-time detection remains a critical area for future research. Optimizing feature extraction processes, exploring parallel computing techniques, and using hardware acceleration can help reduce detection latency. Implementation of these optimizations would be necessary to ensure the practical utility of the model in scenarios where timely detection of phishing threats is of greatest importance.

Future research could explore user-centric approaches to phishing detection that include user behavior analysis and feedback mechanisms. Understanding how users interact with websites and identifying deviations from normal behavior can provide valuable information. Integrating user-centric features into a detection model can improve its adaptability and user-specific threat awareness.

## VII. CONCLUSION

Through the utilization of machine learning algorithms, such as decision trees, support vector machines, and neural networks, the NIDS exhibited commendable accuracy in identifying anomalous activities and distinguishing them from normal network behavior. The ability of these algorithms to adapt and learn from evolving threats makes them valuable assets in the ongoing battle against cyberattacks.

Furthermore, the NIDS showcased efficiency in real-time monitoring, enabling timely response to potential threats and reducing the risk of data breaches. The incorporation of feature engineering and extraction techniques contributed to the system's ability to recognize patterns indicative of malicious intent, enhancing its overall detection capabilities.

Despite these successes, it is crucial to acknowledge that the NIDS is not a foolproof solution. Continuous updates and refinement of the machine learning models are essential to keep pace with the ever-changing landscape of cyber threats. Regular training on new datasets and the incorporation of the latest threat intelligence will further bolster the system's efficacy.

The Network Intrusion Detection System leveraging machine learning techniques represents a promising approach to fortifying network security. Its proactive nature and adaptability make it a valuable component in the defense against a wide array of cyber threats. As technology continues to evolve, so must our cybersecurity measures, and the integration of machine learning in NIDS is a significant stride towards a more robust and resilient network defense strategy.

## REFERENCES

- [1] Wenke Lee and Salvatore J. Stolfo. Data Mining Approaches for Intrusion Detection. In *Proceedings of the 7th USENIX Security Symposium*, 1998.
- [2] M. Tavallaei, E. Bagheri, W. Lu, and A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.
- [3] R. Sommer and V. Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, 2010.
- [4] S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using ensemble of soft computing paradigms. *Journal of Network and Computer Applications*, 30(1), 2007.
- [5] I. Kotenko, R. Yusupov, and A. Chechulin. Application of machine learning methods for intrusion detection system. In *Proceedings of the 9th International Conference for Young Computer Scientists*, 2008.
- [6] R. A. Carrera, T. F. Lunt, and J. M. McDermott. A learning-based approach to the detection of SQL attacks. In *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*, 2005.
- [7] X. Wang, D. Tao, and X. Yang. A novel approach to intrusion detection using artificial neural networks and support vector machines. In *Proceedings of the International Joint Conference on Neural Networks*, 2006.
- [8] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA)*, 2001.
- [9] G. Gu, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the USENIX Security Symposium*, 2008.
- [10] M. I. Heywood, C. D. Nugent, H. Fujita, and J. C. Bezdek. A clustering approach for the optimisation of anomaly detection systems. *Journal of Systems and Software*, 83(5), 2010.
- [11] A. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In *Proceedings of the 8th Annual Computer Security Applications Conference*, 1992.
- [12] J. L. Bentley and M. A. T. Heath. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 1992.
- [13] R. K. Cunningham, M. A. T. Heath, and D. J. Israel. An ensemble learning approach to cyber security intrusion detection. In *Proceedings of the International Joint Conference on Neural Networks*, 2005.
- [14] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Bayesian event classification for intrusion detection. In *Proceedings of the 13th USENIX Security Symposium*, 2004.
- [15] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), 2009.
- [16] Y. Zhang, D. E. Bakken, and B. H. C. Cheng. Anomaly detection in computer security: A statistical machine learning approach. In *Proceedings of the International Joint Conference on Neural Networks*, 2003.