



Kammavari Sangham (R) 1952, K.S.Group of Institutions

K. S INSTITUTE OF TECHNOLOGY

9 No.14, Raghuvanahalli, Kanakapura Road, Bengaluru - 560109

Affiliated to VTU, Belagavi & Approved by AICTE, New Delhi, Accredited by NBA , NAAC & IEI

Department of Computer Science and Engineering Project Phase – 2 (18CSP77)

Project Title: Machine Learning based Network intrusion detection system

Group No.: A1

1KS20CS056 Moniesh S
1KS20CS057 Monika N
1KS20CS073 Pavithra R
1KS20CS093 Sindhura H

Batch No.: 2023_CSE_09

Guided By:
Mr. Somasekhar T
Associate Professor
Dept. of CSE , KSIT

CONTENTS

- Introduction
- Comparison with similar work
- Problem Statement and Objectives
- Methodology
- Technology/Tools used
- Implementation of module with codes
- Snapshots
- References

Introduction

- With more internet-connected devices and increased online activity, it's harder to keep sensitive information and critical systems safe from cyber threats.
- In response to these challenges, Machine Learning (ML) has emerged as a powerful tool to detect and respond to threats, improving overall security.
- ML algorithms are great at finding patterns and unusual things in big sets of data. This helps them find small signs of bad actions that normal security might miss.
- By using ML algorithms, the proposed IDS examines network traffic, checks URLs, and looks at user interactions to stop potential threats before they cause harm.
- Our goal is to protect digital assets and maintain the integrity of online systems.

Comparison with similar work

	Title	Author	Methodology	Outcomes
1	Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network-2019	Hong Yu Yang	The paper proposes an improved Convolutional Neural Network (CNN) for wireless network intrusion detection. The CNN is customized for the specific characteristics of wireless networks.	The improved CNN demonstrates enhanced accuracy in detecting intrusions in wireless networks compared to traditional methods.
2	Generalized Intrusion Detection Mechanism for Empowered Intruders in Wireless Sensor Networks-2020	Wenming Wang	The paper presents a generalized intrusion detection mechanism designed to identify empowered intruders in wireless sensor networks. The mechanism takes into account various types of intrusions that can occur in these networks.	The proposed mechanism shows effectiveness in detecting and mitigating intrusions by empowered attackers within wireless sensor networks.

Comparison with similar work

	Title	Author	Methodology	Outcomes
3	Method of Network Intrusion Discovery Based on Convolutional Long-Short Term Memory Network and Implementation in VSS-2021	Zhijie Fan	This paper introduces a network intrusion detection method based on a Convolutional Long-Short Term Memory (CLSTM) network. The method is implemented in a Virtual Security System (VSS) environment.	The CLSTM network demonstrates improved capabilities in capturing temporal dependencies, leading to enhanced detection accuracy in network intrusions when implemented in the VSS.
4	Effective algorithms to detect stepping-stone intrusion by removing outliers of packet RTTs-2022	Lixin Wang	The paper proposes algorithms to detect stepping-stone intrusions by analyzing packet Round-Trip Times (RTTs) and removing outliers. The focus is on identifying anomalous patterns indicative of stepping-stone attacks.	The proposed algorithms effectively identify and mitigate stepping-stone intrusions by filtering out outlier RTT values, improving the overall security of the network.

Problem Statement and Objectives

Problem Statement:

The increasing sophistication of malicious attacks poses a significant challenge to conventional security measures. One critical area of concern is the propagation of malicious links, which can lead to data breaches and unauthorized access to sensitive information. Current IDS often lack proactive mechanisms to verify the safety of links and mitigate potential risks effectively .

Objectives:

- Integrate machine learning algorithms to accurately differentiate between authentic and phishing websites.
- Develop an open-source system for extracting features and maintaining an up-to-date dataset of phishing and genuine websites.
- Reduce the reliance on a vast number of features by selecting key characteristics supported by evidence.
- Mitigate dataset bias by ensuring a balanced representation of URL and content-based attributes.
- Enhance the ability to protect internet users from phishing attacks by improving the accuracy of phishing detection.

Methodology

Steps involved in implementation :

➤ Dataset Collection:

- Gather a diverse and representative dataset containing labeled instances of normal network behavior and various types of intrusions. Utilize publicly available datasets and, if possible, collaborate with industry partners to ensure realism and relevance.

➤ Data Preprocessing:

- Clean and preprocess the collected dataset by handling missing values, normalizing features, and addressing any inconsistencies. This step is crucial for ensuring the quality and reliability of the data used for training and testing.

➤ Feature Engineering:

- Extract relevant features from the preprocessed data to characterize network traffic patterns effectively. Feature engineering may involve selecting key attributes, transforming variables, and creating new features to enhance the performance of machine learning models.

Methodology

➤ Model Selection:

Evaluate and choose suitable machine learning algorithms for intrusion detection, considering factors such as accuracy, interpretability, and scalability. Common choices include decision trees, random forests, support vector machines, and deep learning models.

➤ Model Training:

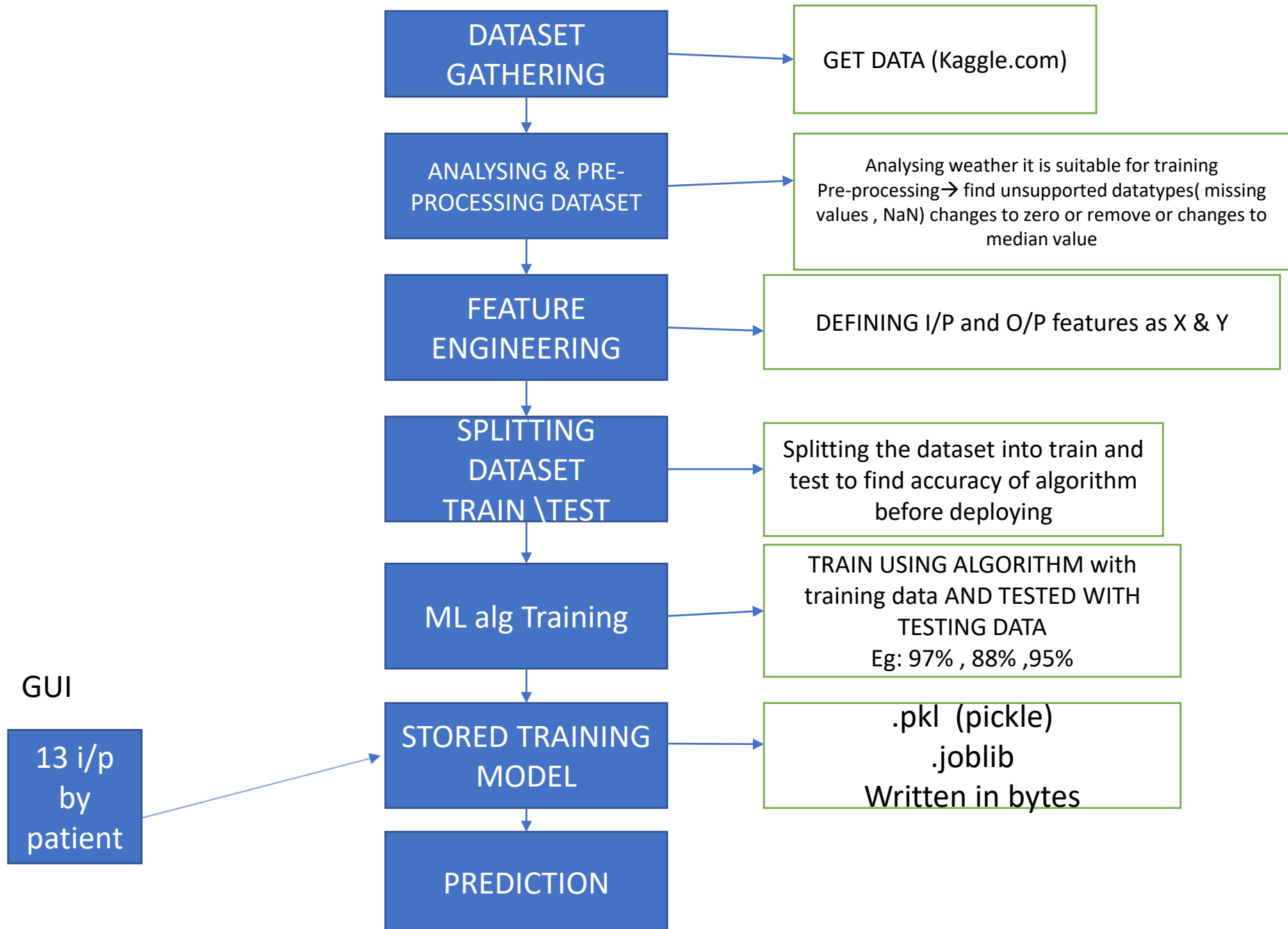
Train the selected machine learning models using the preprocessed and engineered dataset. Employ techniques such as cross-validation to optimize model hyperparameters and ensure robust generalization to unseen data.

➤ Model Evaluation:

Assess the performance of the trained models using separate test datasets, employing metrics such as precision, recall, F1 score, and ROC-AUC. Iterate on the model training and evaluation process to fine-tune the system for optimal results.

➤ User Interface:

Represents the user interacting with the system. The user uploads URLs for analysis, initiating the process of intrusion detection.



Technology/Tools used

- Programming Language : Python 3.6
- Tools : Python IDLE
- Web interface : Flask Web Framework
- Libraries : Scikit-learn & TensorFlow(ML tasks)
- Database : SQLite Database
- KNN for Classification
- BeautifulSoup(web)

Implementation of module with codes

findport.py •

C: > Users > Lenovo > Desktop > hacker > main > main > findport.py > ...

```
1  import socket
2
3  def find_available_port(start_port, end_port):
4      for port in range(start_port, end_port + 1):
5          try:
6              s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7              s.bind(("localhost", port))
8              s.close()
9              return port
10         except OSError as e:
11             if e.errno == 98 or e.errno == 10048: # errno 98: Address already in use
12                 continue
13             else:
14                 raise
15         raise RuntimeError("No available ports in the specified range")
16
17 # Example usage
18 start_port = 8000
19 end_port = 8100
20 port = find_available_port(start_port, end_port)
21 print(f"Found available port: {port}")
22
23 import socket
24 |
25 # Get the hostname of the local machine
26 hostname = socket.gethostname()
27
28 # Get the IP address of the local machine
29 ip_address = socket.gethostbyname(hostname)
30 print(ip_address)
31
```

Ln 24, Col 1

PROBLEMS OUTPUT TERMINAL ...

Python Debug Console + - [] [x] ... ^ x

File "C:\Users\Lenovo\Desktop\hacker\main\main\findport.py", line 24

``

^

SyntaxError: invalid syntax

```
PS C:\Users\Lenovo\Desktop\hacker\main\main> c:; cd 'c:\Users\Lenovo\Desktop\hacker\main\main'; & 'c:\Users\Lenovo\AppData\Local\Programs\Python\Python37\python.exe' 'c:\Users\Lenovo\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '49699' '--' 'C:\Users\Lenovo\Desktop\hacker\main\main\findport.py'
```

Found available port: 8000

127.0.0.1

```
PS C:\Users\Lenovo\Desktop\hacker\main\main>
```

Snapshots

INTRUSION DETECTION [SIGNIN](#) [SIGNUP](#)

User Login

Username:

Password:

[Submit](#)

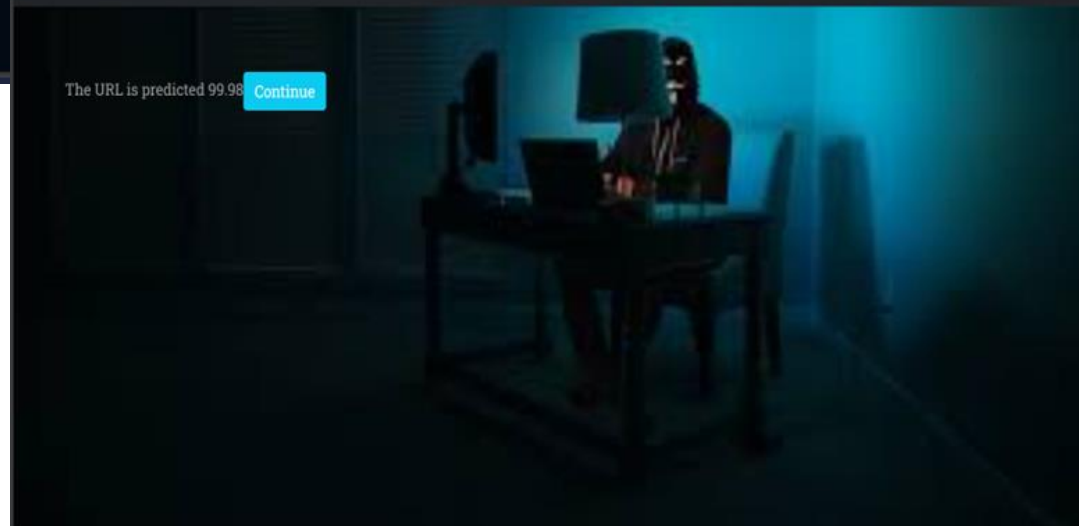
INTRUSION DETECTION [LOGOUT](#)

Send link

[Submit](#)

INTRUSION DETECTION [LOGOUT](#)

The URL is predicted 99.98% [Continue](#)



References

1. Mahmoud, S. A., & Al-Dabbagh, A. M. (2018). A Review of Machine Learning Techniques for Intrusion Detection Systems. *International Journal of Advanced Computer Science and Applications*, 9(10), 134-141.
2. Wang, S. S., & Lu, Y. H. (2020). Machine Learning-Based Intrusion Detection System for Advanced Persistent Threats: A Review. *International Journal of Advanced Computer Science and Applications*, 11(4), 117-124.
3. Choudhary, S., Rani, R., & Kumar, V. (2021). An Intrusion Detection System Based on Machine Learning and Internet of Things. *IEEE Internet of Things Journal*, 8(3), 1625-1633.
4. Kumar, R., Sharma, S., & Jain, V. (2019). Enhancing Cybersecurity with Machine Learning-Based Intrusion Detection Systems: A Survey. *International Journal of Computer Applications*, 182(28), 20-25.
5. Elhag, I. H., Albishi, A. M., & Alabdraba, W. A. (2020). Using Machine Learning Techniques for Intrusion Detection: A Comprehensive Survey. *IEEE Access*, 8, 203498-203518.

Thank You!