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

K. S INSTITUTE OF TECHNOLOGY

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 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
- Results
- Future Work
- Conclusion
- References

Introduction

- There are 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 detect digital assets and maintain the integrity of online systems.

Literature Survey

	Title	Author	Methodology	Outcomes
1.	Learning From Few Cyber-Attacks: Addressing the Class Imbalance Problem in Machine Learning-Based Intrusion Detection in Software-Defined Networking -2023	Zhiwei Qin	The paper proposes a method to address the class imbalance problem in machine learning-based intrusion detection systems (IDS) deployed in Software-Defined Networking (SDN) environments. It utilizes a few-shot learning approach to learn from limited attack samples and improve the detection performance of the IDS.	The paper's method likely led to higher accuracy and robustness in intrusion detection within Software-Defined Networking environments by effectively addressing the class imbalance problem and leveraging few-shot learning techniques.
2.	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.

Literature Survey

	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.	Generalized Intrusion Detection Mechanism for Empowered Intruders in Wireless Sensor Networks-2021	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.

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 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 detect internet users from malicious attacks by improving the accuracy of intrusion 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:**

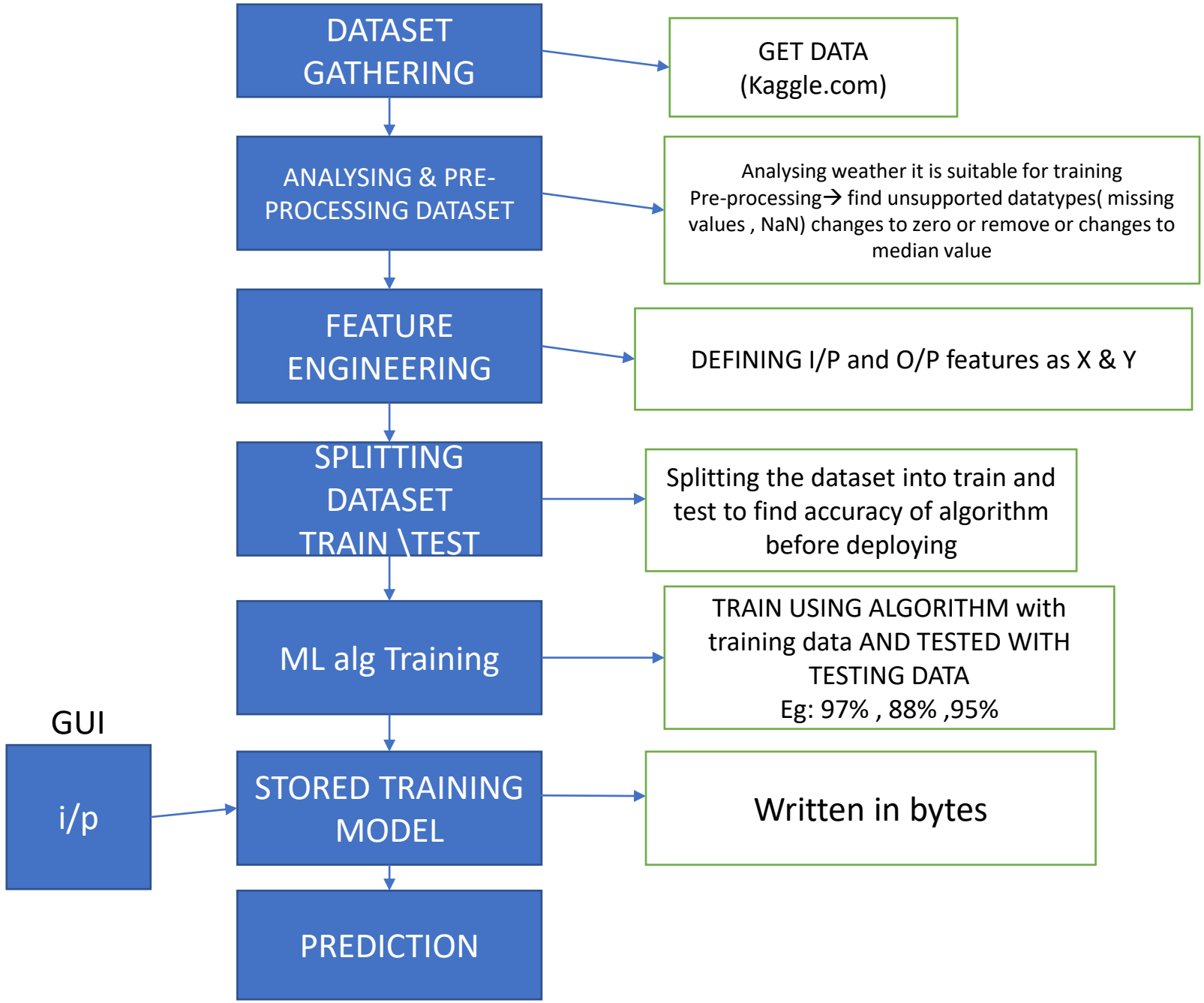
  Train the selected machine learning models using the preprocessed and engineered dataset. Employ techniques such as cross-validation to optimize model hyperparameters and ensure robust generalization to unseen data.

- **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.


DATASET GATHERING
GET DATA (Kaggle.com)
ANALYSING & PRE-PROCESSING DATASET
Analysing weather it is suitable for training
Pre-processing→ find unsupported datatypes(missing values , NaN) changes to zero or remove or changes to median value
FEATURE ENGINEERING
DEFINING I/P and O/P features as X & Y
SPLITTING DATASET TRAIN \TEST
Splitting the dataset into train and test to find accuracy of algorithm before deploying
ML alg Training
TRAIN USING ALGORITHM with training data AND TESTED WITH TESTING DATA
Eg: 97% , 88% ,95%
GUI
i/p
STORED TRAINING MODEL
Written in bytes
PREDICTION

Technology/Tools used

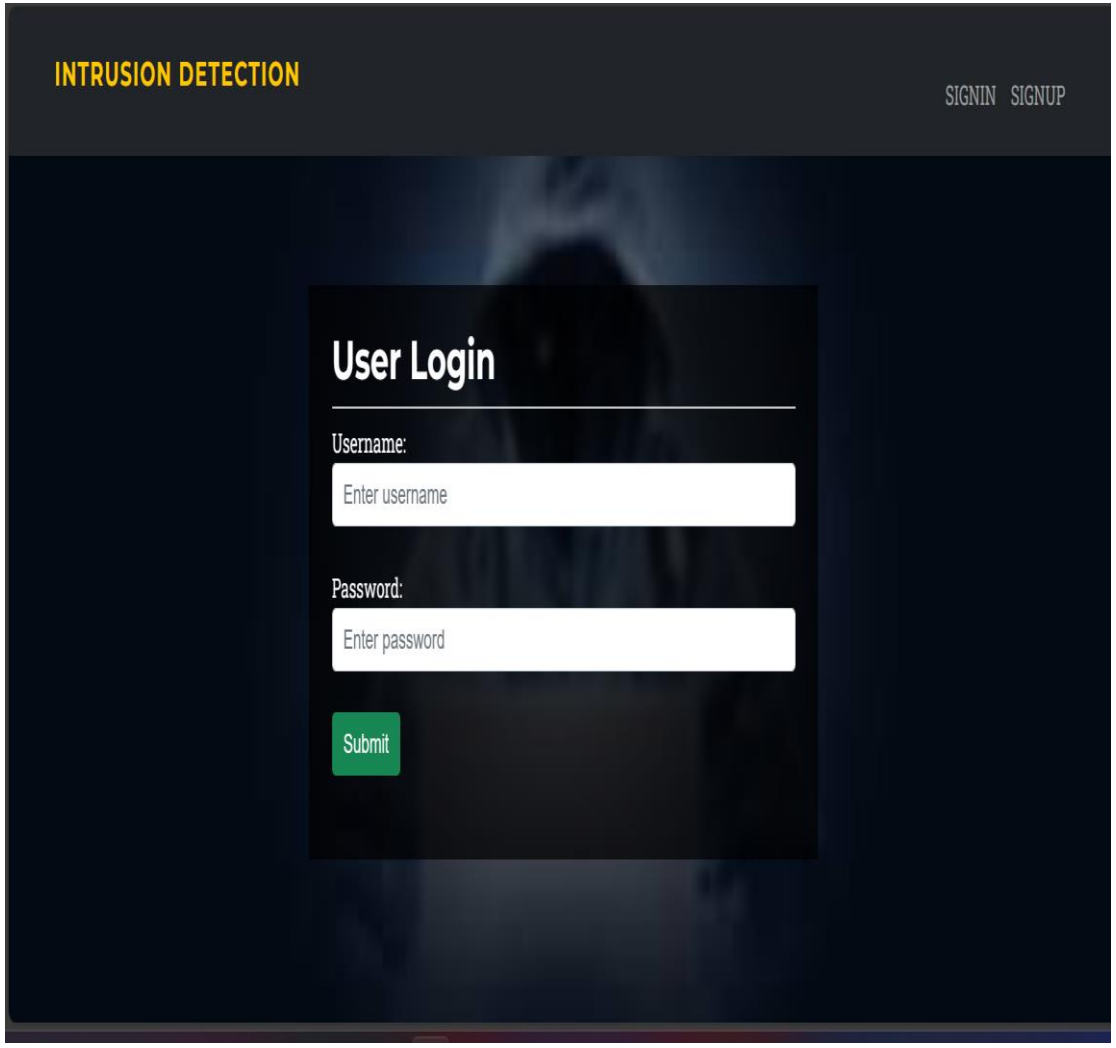
- Programming Language : Python 3.7.1
- Tools : Python IDLE
- Web interface : Flask Web Framework
- Libraries : Scikit-learn & TensorFlow(ML tasks)
- Database : SQLite Database
- KNN for Classification
- Beautifulsoup4

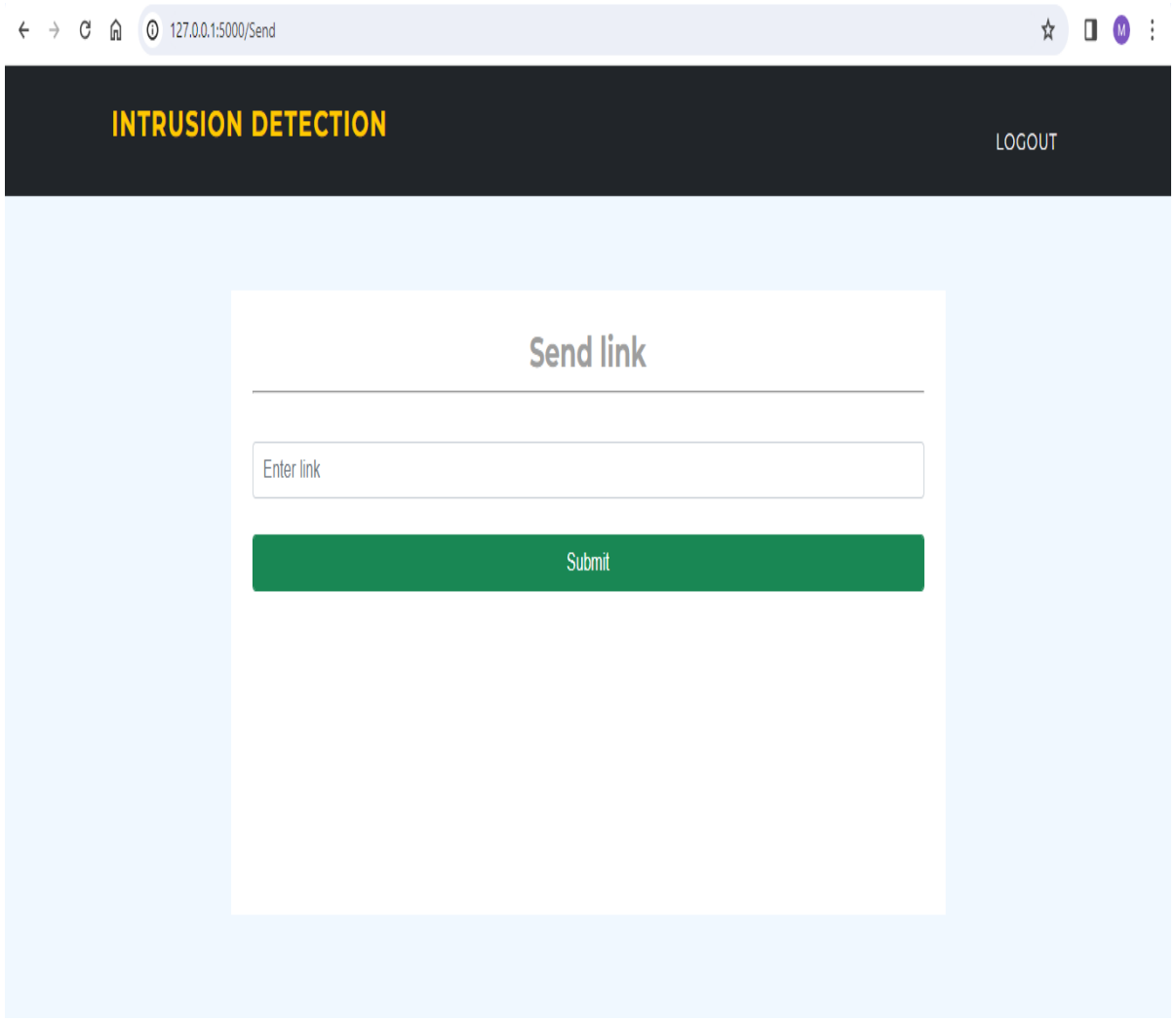
Implementation of module with codes

```
from flask import Flask, render_template, url_for, request, jsonify, session
import sqlite3
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings('ignore')
from keras.models import load_model
from feature import FeatureExtraction
import os
model = load_model('model/model.h5')


file = open("model/model.pkl","rb")
gbc = pickle.load(file)
file.close()


connection = sqlite3.connect('user_data.db')
cursor = connection.cursor()

command = """CREATE TABLE IF NOT EXISTS user(name TEXT, password TEXT, mobile TEXT, email TEXT)"""
cursor.execute(command)

app = Flask(__name__)
app.secret_key = os.urandom(24)

def getData():
    while True:
        try:
            data = client_socket.recv(1024)
            break
        except:
            print("something went wrong")
    return data

@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/userlog', methods=['GET', 'POST'])
def userlog():
    if request.method == 'POST':

        connection = sqlite3.connect('user_data.db')
        cursor = connection.cursor()

        name = request.form['name']
        password = request.form['password']

        query = "SELECT * FROM user WHERE name = '"+name+"' AND password= '"+password+"'"
        cursor.execute(query)

        result = cursor.fetchone()

        if result:
            session['user'] = result
            return render_template('userlog.html')
        else:
            return render_template('index.html', msg='Sorry, Incorrect Credentials Provided,  Try Again')

    return render_template('index.html')

@app.route('/userreg', methods=['GET', 'POST'])
def userreg():
    if request.method == 'POST':

        connection = sqlite3.connect('user_data.db')
        cursor = connection.cursor()

        name = request.form['name']
        password = request.form['password']
        mobile = request.form['phone']
        email = request.form['email']

        print(name, mobile, email, password)

        command = """CREATE TABLE IF NOT EXISTS user(name TEXT, password TEXT, mobile TEXT, email TEXT)"""
        cursor.execute(command)
```

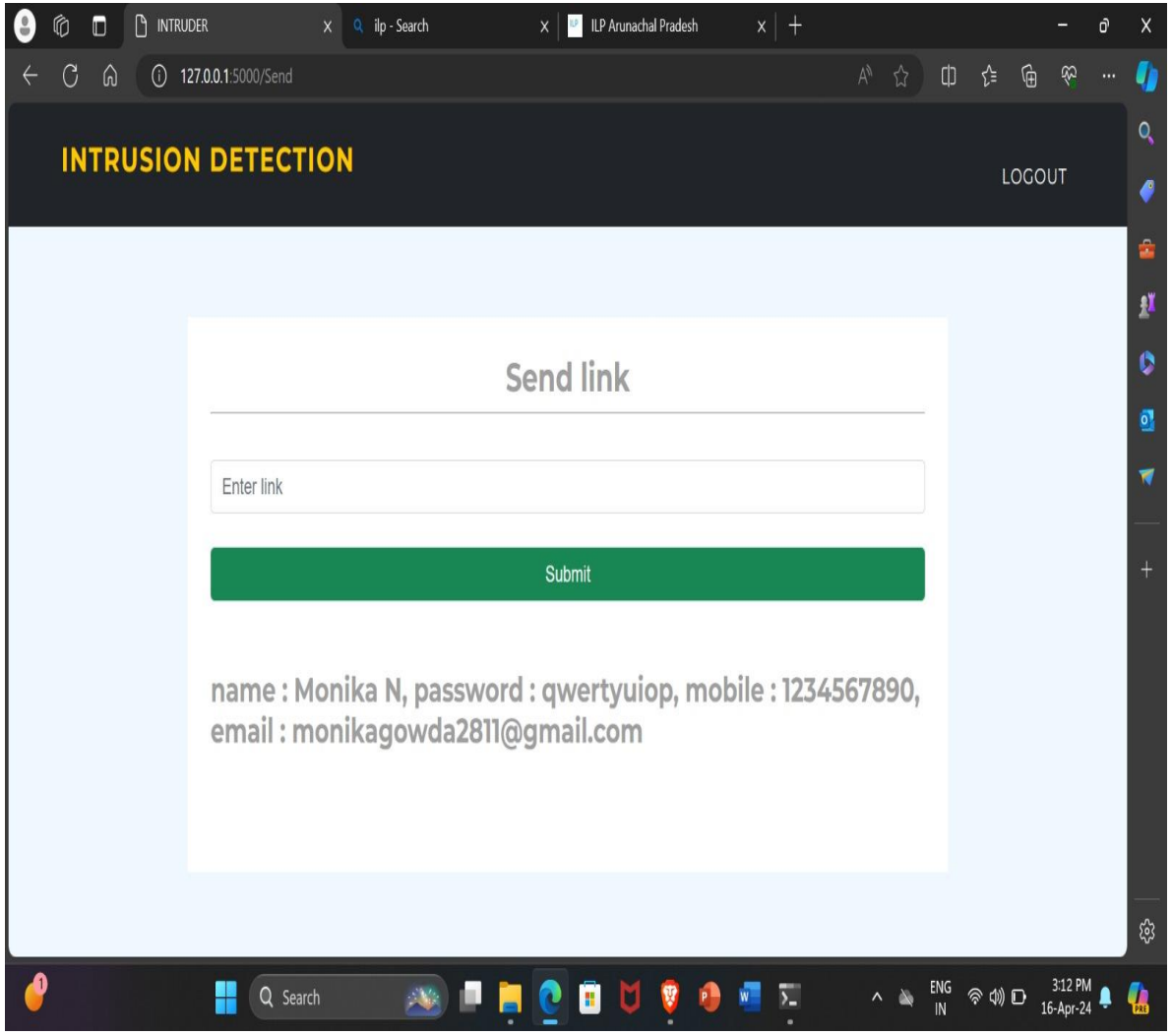
Results


INTRUSION DETECTION
SIGNIN SIGNUP
User Login
Username:
Enter username
Password:
Enter password
Submit


127.0.0.1:5000/Send
INTRUSION DETECTION
LOGOUT
Send link
Enter link
Submit

Results


INTRUSION DETECTION
LOGOUT
The URL is predicted 99.98
Continue


INTRUDER
ilp - Search
ILP Arunachal Pradesh
127.0.0.1:5000/Send
INTRUSION DETECTION
LOGOUT
Send link
Enter link
Submit
name : Monika N, password : qwertyuiop, mobile : 1234567890,
email : monikagowda2811@gmail.com
Search
ENG
IN
3:12 PM
16-Apr-24

Future Work

Leveraging Machine Learning and AI

- Employ AIML algorithms to continuously improve the effectiveness.

Implementing Security Measures

- Enhance the security posture of detected malicious websites through the implementation of robust security protocols and mechanisms.
- This may include deploying encryption, secure authentication mechanisms, and ensuring data integrity through techniques such as digital signatures and hash functions.

Utilizing Web Application Firewalls (WAF)

- Integrate WAF technology to provide an additional layer of defense against web-based attacks. WAFs can inspect and filter HTTP traffic to detect and block malicious requests.

Conclusion

Enhanced Website Security

- Implementation of robust security measures, including web application firewalls, content security policies, and DNS-based security solutions, significantly enhances the security posture of detected malicious websites.

Proactive Threat Mitigation

- Integration of threat intelligence feeds and behavioral analysis enables proactive detection and blocking of malicious activities, reducing the risk of successful cyber attacks.

Continuous Improvement

- Leveraging machine learning and artificial intelligence facilitates continuous improvement of the security solution, allowing for adaptive defenses that can effectively counter emerging cyber threats.

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!