

Project School

Project Title: NETWORK INTRUSION DETECTION SYSTEM

Faculty Incharge: Dr. Rajasekaran.

Session Duration: 10-09-2022 to 17-12-2022



Name: Shubham Mola.

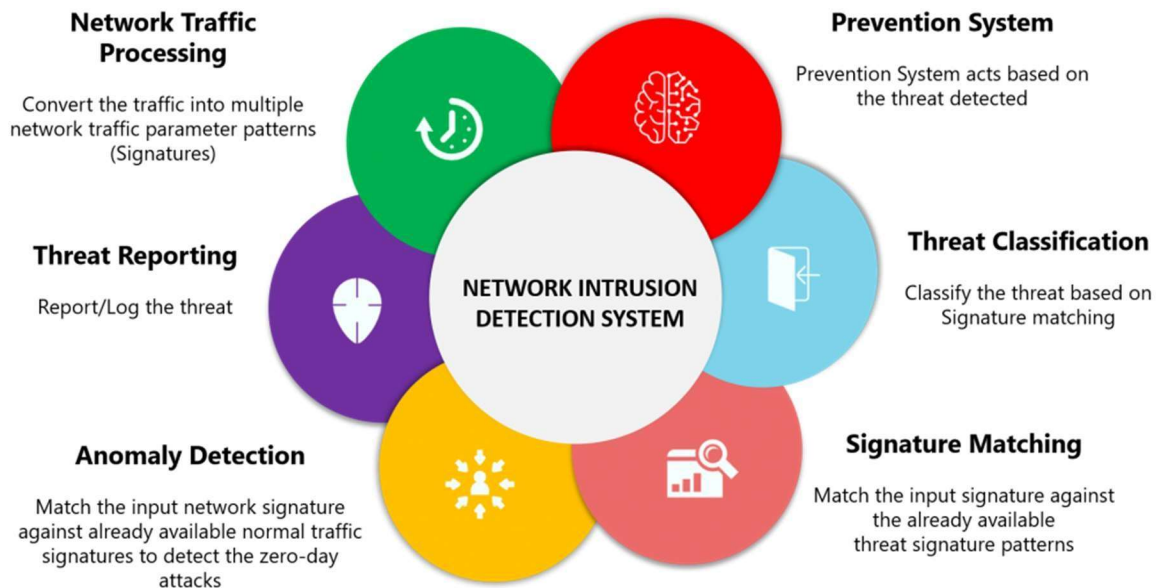
Roll No:20BD1A057H.

Class: CSE-D.

Signature of student

Signature of Mentor.

Domain:



Project Description:

This web application helps to identify an attack or sense abnormal behavior in the network and send an alert to the user and protect the user. When the user login into the portal he gets the information about the network's accuracy, f1-score, precision. This helps the user to detect how safe his network is for the system.

Introduction:

A network intrusion refers to any forcible or unauthorized activity on a digital network. Attacks can broadly classified into four major categories:

- DOS(Denial of Service)
- Probe
- R2L(Remote-to-User)
- U2R(User-to-Root)

Network intrusion detection systems are placed at a strategic point within the network to examine traffic from all devices on the network. Once it identifies an attack or senses abnormal behavior, it sends an alert to the user.

Technologies Used:

1. Model Building:

- Preprocessing: LabelEncoder (in sklearn.preprocessing)
- Normalization: MinMaxScalar (in sklearn.preprocessing)

2. Web Integration:

- Stack Selected: MERN (MongoDB, Express, React ,Node)

3. Advanced Features:

Using Google Earth 2.0 for user authentication and storing user details in a hash in MongoDB.

4. Database:

Data is stored in MongoDB and can be imported and exported. The data is safe and secured.

5. Backend :

The backend is developed using Python.

Dataset Used:

NSL-KDD

- NSL-KDD is a new version of the KDD'99 data set.
- This is an effective benchmark data set to help researchers compare different intrusion detection methods.

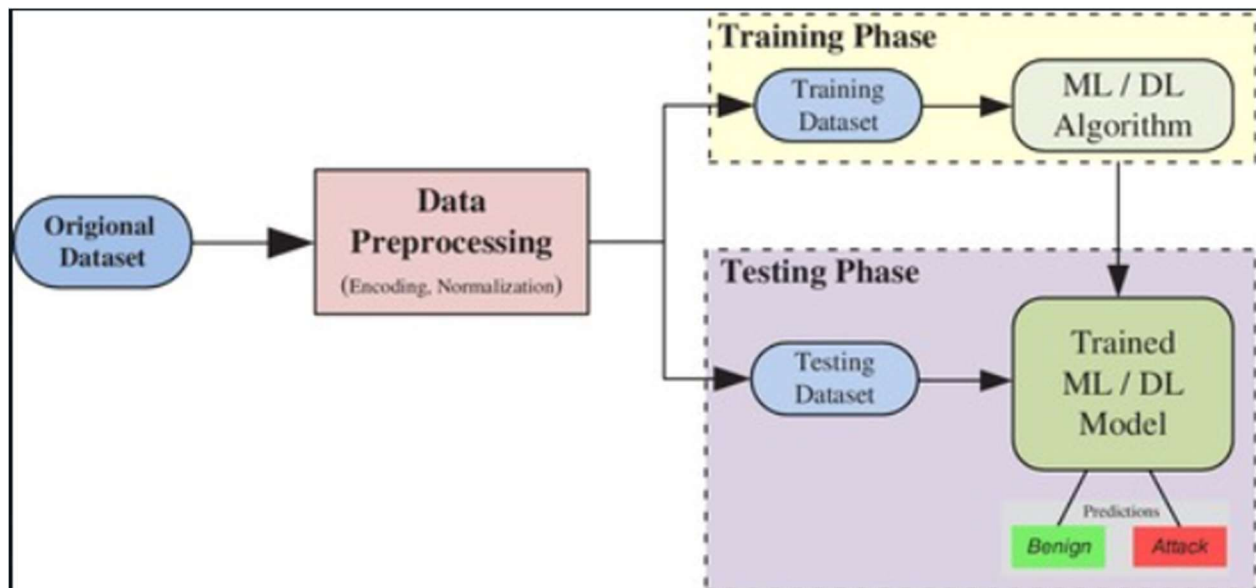
Data files:

- KDDTrain+.ARFF: The full NSL-KDD train set with binary labels in ARFF format.
- KDDTrain+.TXT: The full NSL-KDD train set including attack-type labels and difficulty level in CSV format.
- KDDTrain+_20Percent.ARFF: A 20% subset of the KDDTrain+.arff file.
- KDDTrain+_20Percent.TXT: A 20% subset of the KDDTrain+.txt file.
- KDDTest+.ARFF: The full NSL-KDD test set with binary labels in ARFF format.
- KDDTest+.TXT: The full NSL-KDD test set including attack-type labels and difficulty level in CSV format.
- KDDTest-21.ARFF: A subset of the KDDTest+.arff file which does not include records with a difficulty level of 21 out of 21.
- KDDTest-21.TXT: A subset of the KDDTest+.txt file which does not include records with a difficulty level of 21 out of 21.

Algorithms:

- KNN - K-Nearest Neighbors is one of the simplest Machine Learning algorithms. The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
- Random Forest - Random Forest is a popular machine learning algorithm. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
- CNN - Convolutional Neural Networks are neural networks that share their parameters. It is a Deep Learning Algorithm.
- LSTM - Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step.

Process Flow:





Code:

```

1 > import numpy as np
2 import sys
3 from sklearn.metrics import accuracy_score, confusion_matrix
4 import pandas as pd
5 from sklearn.preprocessing import LabelEncoder
6 from sklearn.preprocessing import MinMaxScaler
7 import sklearn
8 from sklearn.neighbors import KNeighborsClassifier
9 import os
10 from sklearn.preprocessing import LabelEncoder
11 import tensorflow as tf
12 from sklearn.preprocessing import Normalizer
13 import pickle
14 data_validate=pd.read_csv('fs_new_validation_project.csv')
15 columns = ([ 'protocol_type', 'service', 'flag', 'logged_in', 'count', 'srv_error_rate', 'srv_reconn_rate', 'same_srv_rate', 'diff_srv_rate', 'dst_host_co
16 data_validate.columns=columns
17 protocol_type = LabelEncoder()
18 service_le = LabelEncoder()
19 flag_le = LabelEncoder()
20 data_validate['protocol_type'] = protocol_type.le.fit_transform(data_validate['protocol_type'])
21 data_validate['service'] = service_le.fit_transform(data_validate['service'])
22 data_validate['flag'] = flag_le.fit_transform(data_validate['flag'])
23 df_validate=data_validate.copy(deep=True)
24 x_validate=df_validate.drop(['!attack'],axis=1)
25 y_validate=pd.DataFrame(df_validate['!attack'])
26 label_encoder = LabelEncoder()
27 scaler=MinMaxScaler()
28 x1=x_validate.copy(deep=True)
29 scaler=MinMaxScaler()
30 scaler.fit(x1)
31 scaled_data=scaler.transform(x1)
32 scaled_data=pd.DataFrame(scaled_data)

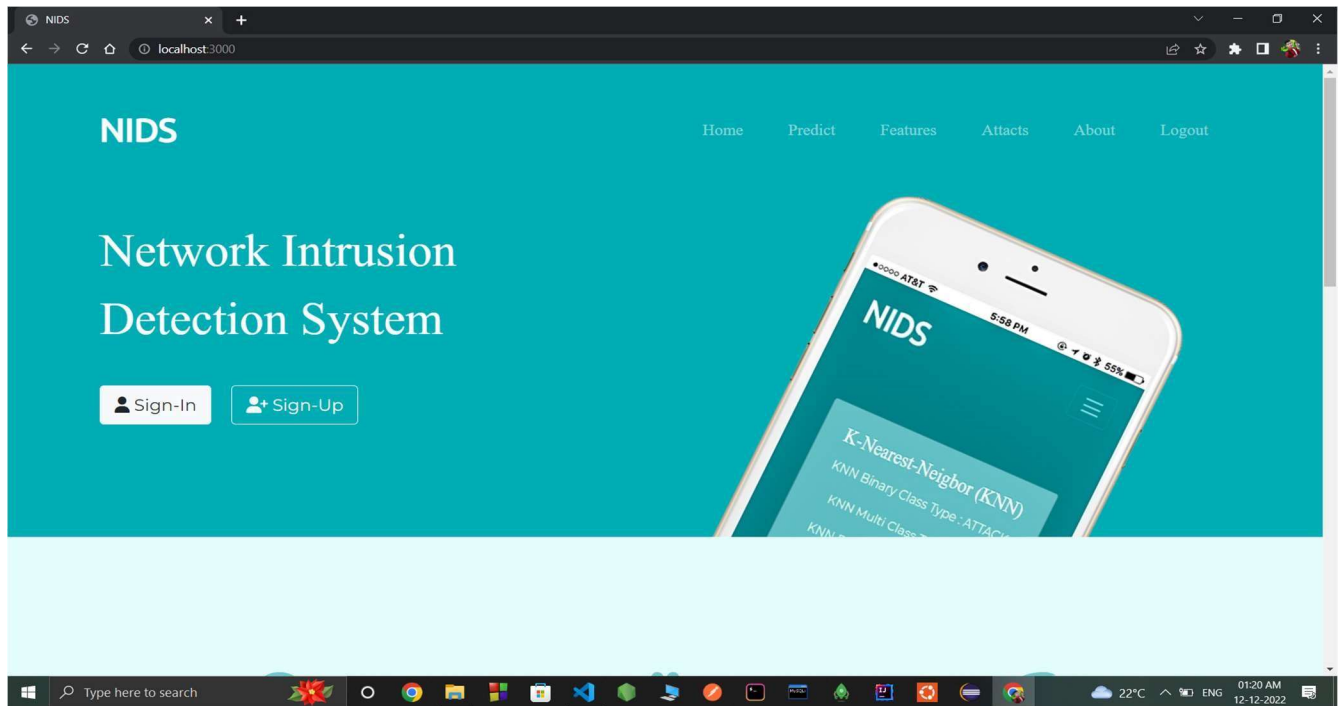
```

```

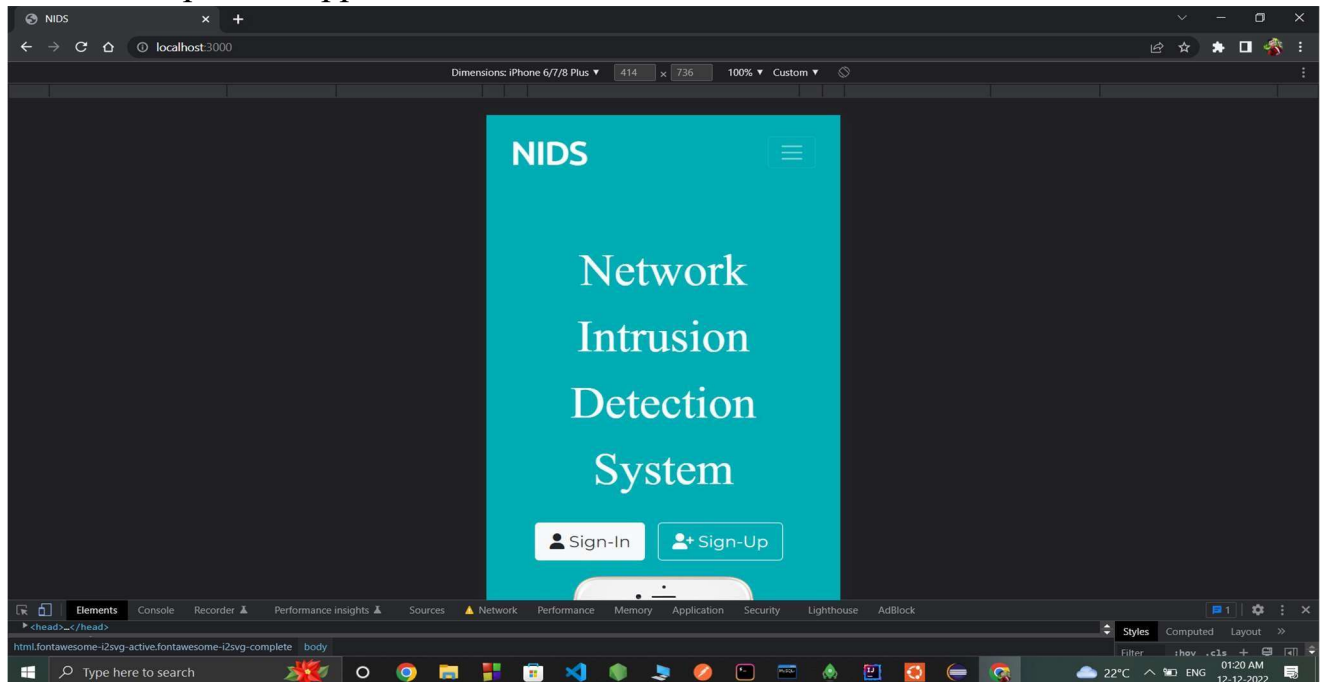
11 = preprocessing.Normalizer()
12 knn_bin = pickle.load(open('knn_binary_class.sav', 'rb'))
13 knn_multi = pickle.load(open('knn_multi_class.sav', 'rb'))
14 randfor_bin = pickle.load(open('random_forest_binary_class.sav', 'rb'))
15 randfor_multi = pickle.load(open('random_forest_multi_class.sav', 'rb'))
16 cmn_bin = tf.keras.models.load_model('latest_cmn_bin.h5')
17 cmn_multi = tf.keras.models.load_model('latest_cmn_multiclass.h5')
18 lstm_bin = tf.keras.models.load_model('lstm_latest_bin.h5')
19 lstm_multi = tf.keras.models.load_model('lstm_latest_multiclass.h5')
20 val_knn_knn_bin.predict(11)
21 if(val_knn[0]==0):
22     print('KNN Algorithm binary class:Normal')
23 else:
24     print('KNN Algorithm binary class:Attack')
25 #print('KNN Algorithm multi class:',knn_multi.predict(11))
26 tp_knn=cmn_multi.predict(11)
27 for i in tp_knn:
28     tp_knn=i
29
30 print('KNN Multi Class Type : ',tp_knn)
31
32 if(tp_knn== 'dos'):
33     print('KNN Description : A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible.'tp_knn='probe')
34     print('KNN Description : Probing is another type of attack in which the intruder scans network devices to determine weakness in topology.'tp_knn='r2l')
35     print('KNN Description : Remote to user (R2U) is a type of computer network attacks, in which an intruder sends set of packets to a user.'tp_knn='u2r')
36     print('KNN Description : User to root attacks (U2R) is an another type of attack where the intruder tries to access the network resources.'
37 else:
38     print('KNN Description : Data is safe')
39 val_rnd=randfor_bin.predict(11)
40 if(val_rnd[0]==0):

```

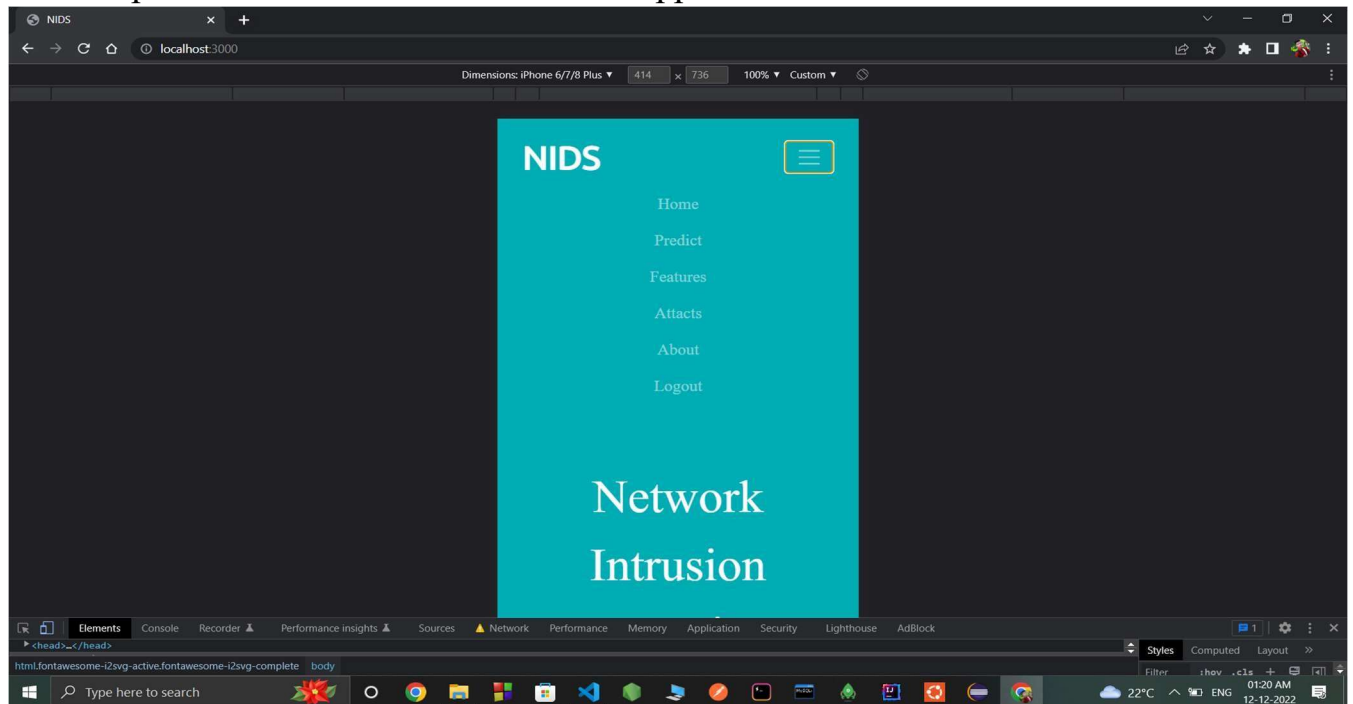
Desktop/Laptop Compatible Application screen:



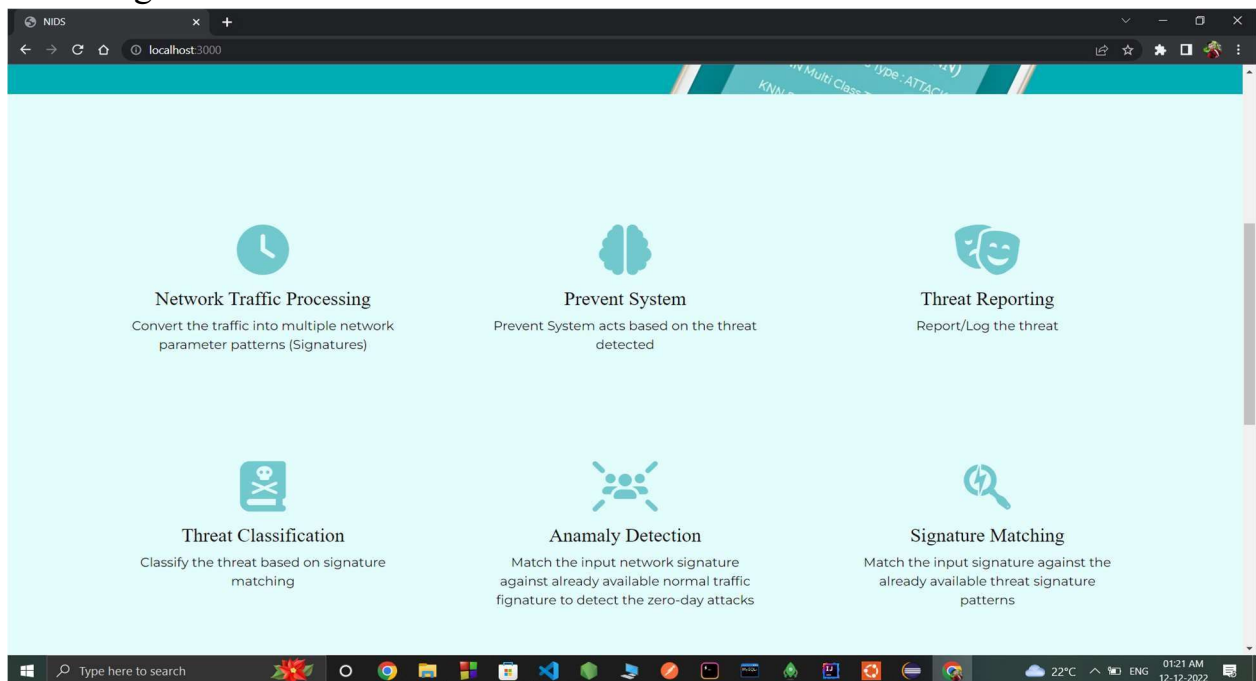
Mobile Compatible Application Screen



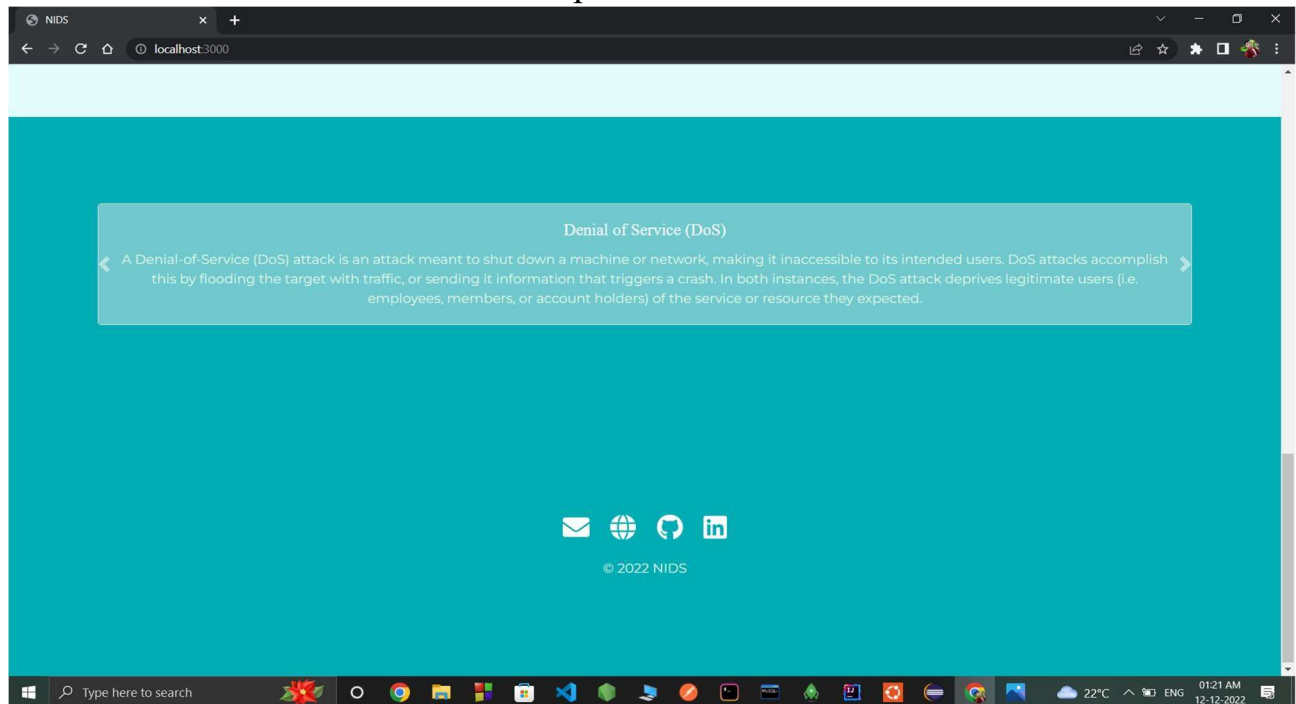
Services provided with a button on mobile app



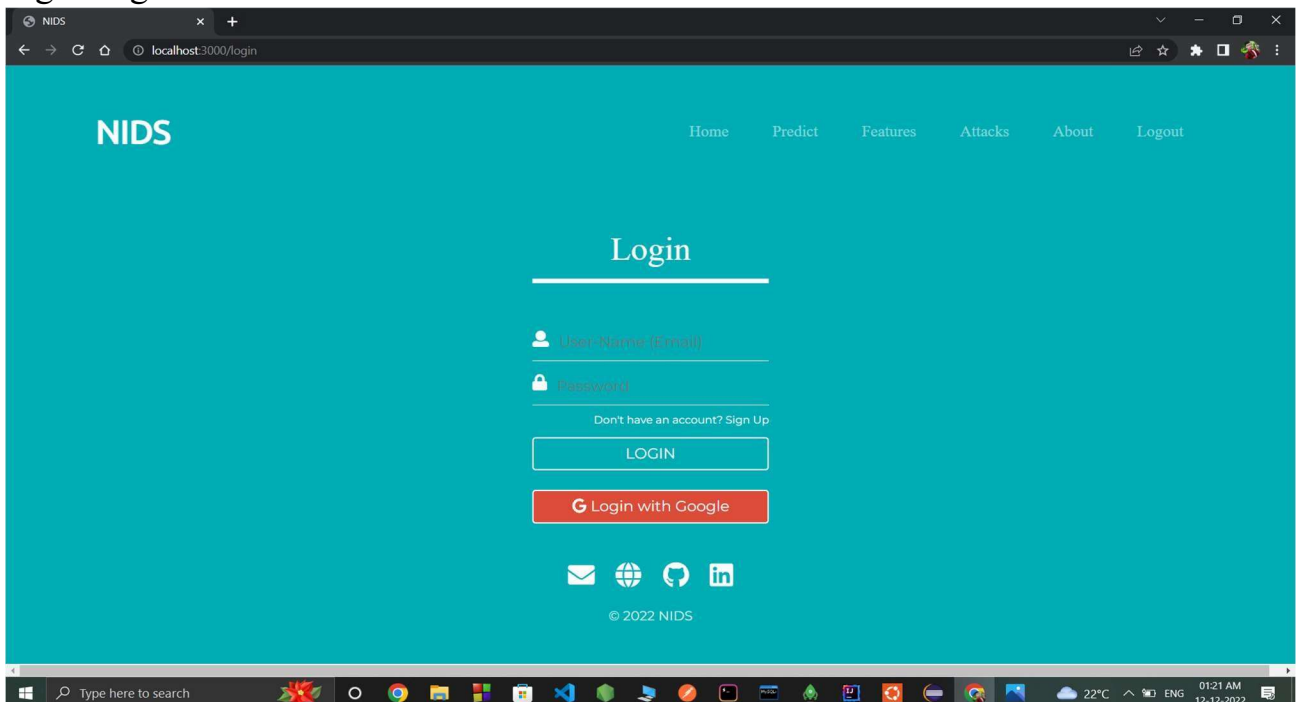
Home Page



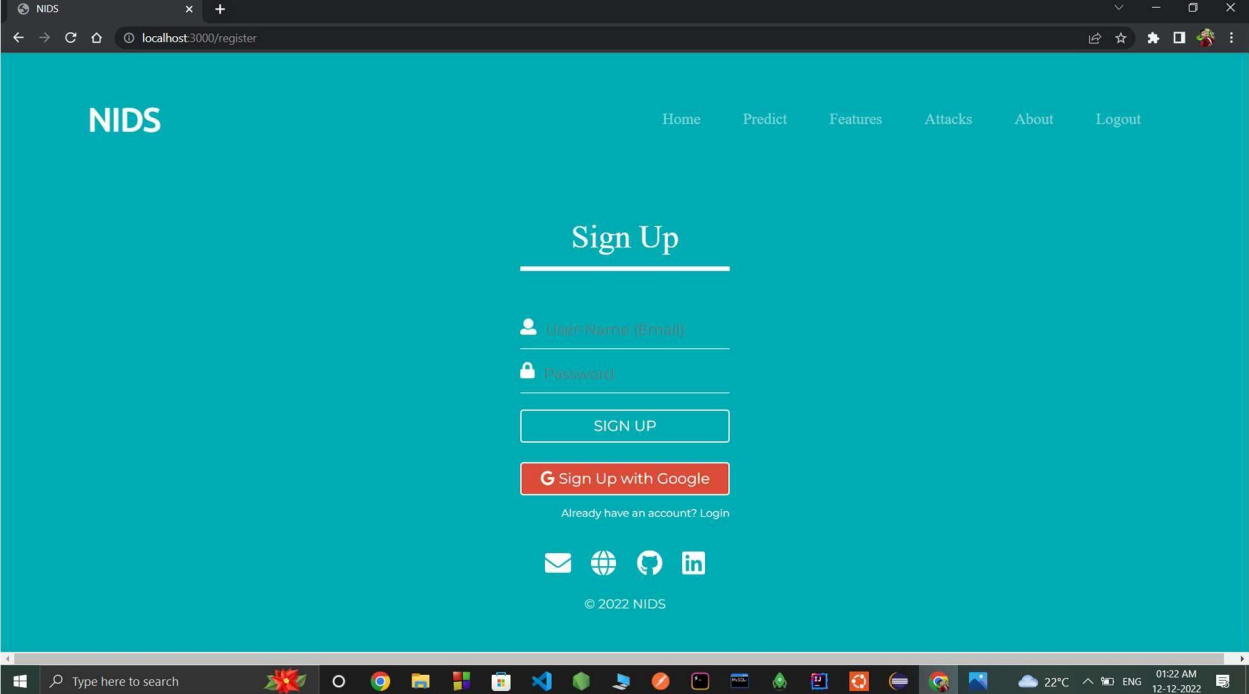
Information of each attack with the help Carousel



Login Page



Sign Up Page



The screenshot shows a web browser window with the URL `localhost:3000/register`. The page has a teal background and a navigation bar at the top with links: Home, Predict, Features, Attacks, About, and Logout. The main heading is "Sign Up". Below it are two input fields: "User Name (Email)" and "Password". A "SIGN UP" button is positioned below the password field. Below the button is a red "Sign Up with Google" button. A link "Already have an account? Login" is located below the Google button. At the bottom, there are four social media icons (Email, Facebook, GitHub, LinkedIn) and a copyright notice "© 2022 NIDS". The Windows taskbar at the bottom shows the time as 01:22 AM on 12-12-2022.

NIDS

Home Predict Features Attacks About Logout

Sign Up

User Name (Email)

Password

SIGN UP

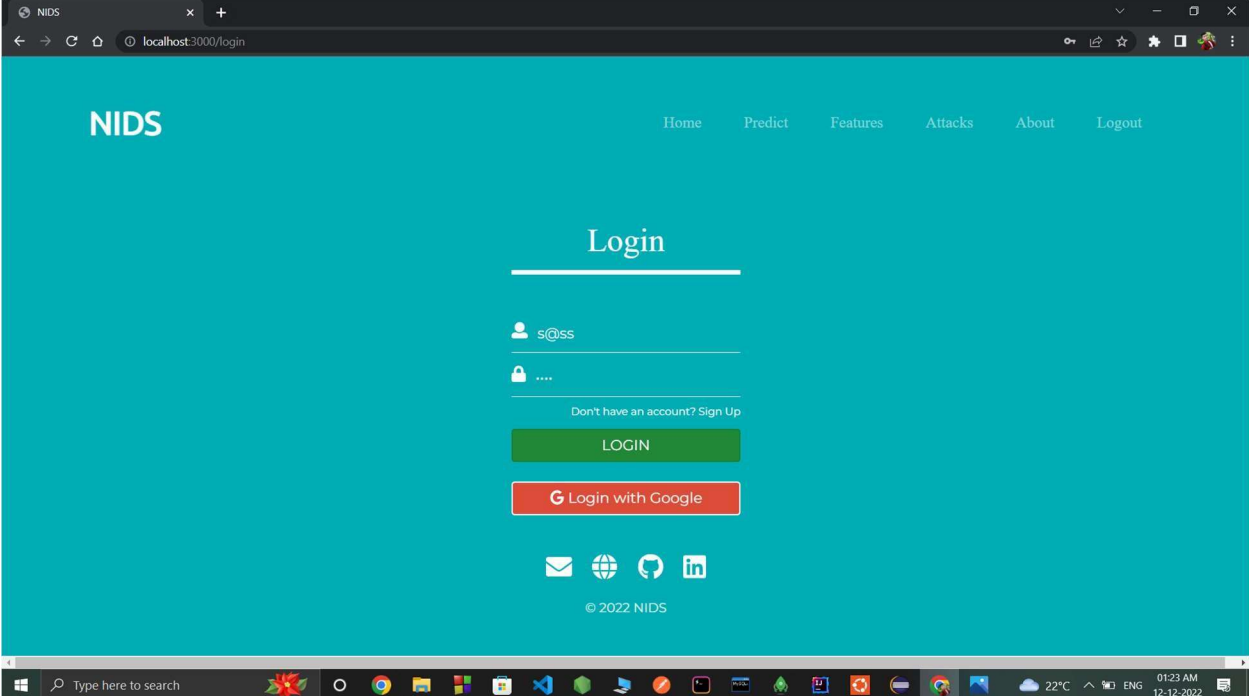
Sign Up with Google

Already have an account? Login

Email Facebook GitHub LinkedIn

© 2022 NIDS

User enter details and validation process



The screenshot shows a web browser window with the URL `localhost:3000/login`. The page has a teal background and a navigation bar at the top with links: Home, Predict, Features, Attacks, About, and Logout. The main heading is "Login". Below it are two input fields: "s@ss" and "****". A "LOGIN" button is positioned below the password field. Below the button is a red "Login with Google" button. A link "Don't have an account? Sign Up" is located below the Google button. At the bottom, there are four social media icons (Email, Facebook, GitHub, LinkedIn) and a copyright notice "© 2022 NIDS". The Windows taskbar at the bottom shows the time as 01:23 AM on 12-12-2022.

NIDS

Home Predict Features Attacks About Logout

Login

s@ss

LOGIN

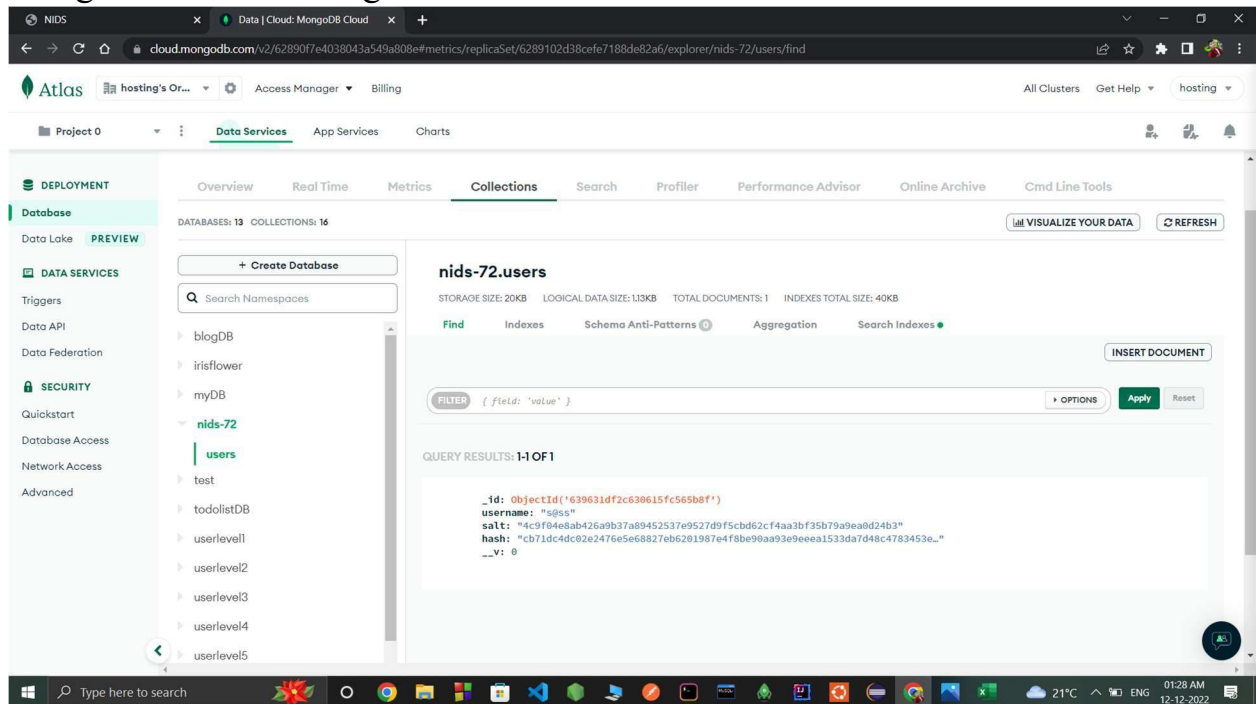
Login with Google

Don't have an account? Sign Up

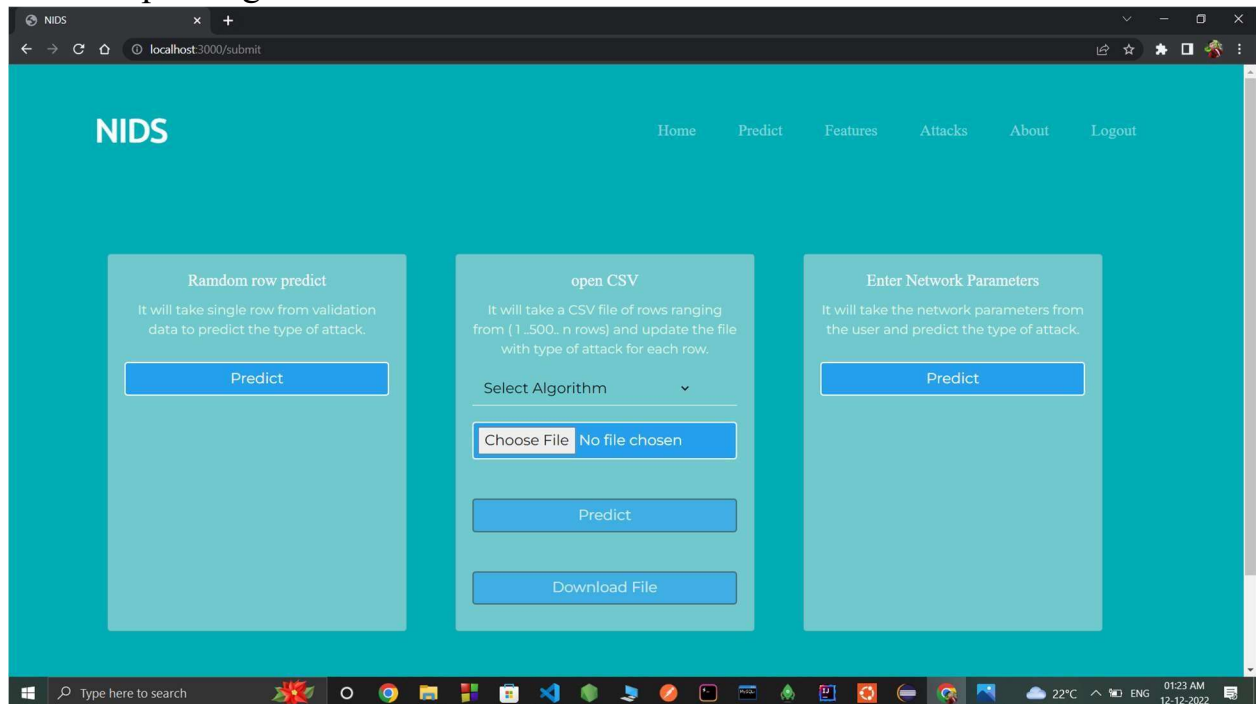
Email Facebook GitHub LinkedIn

© 2022 NIDS

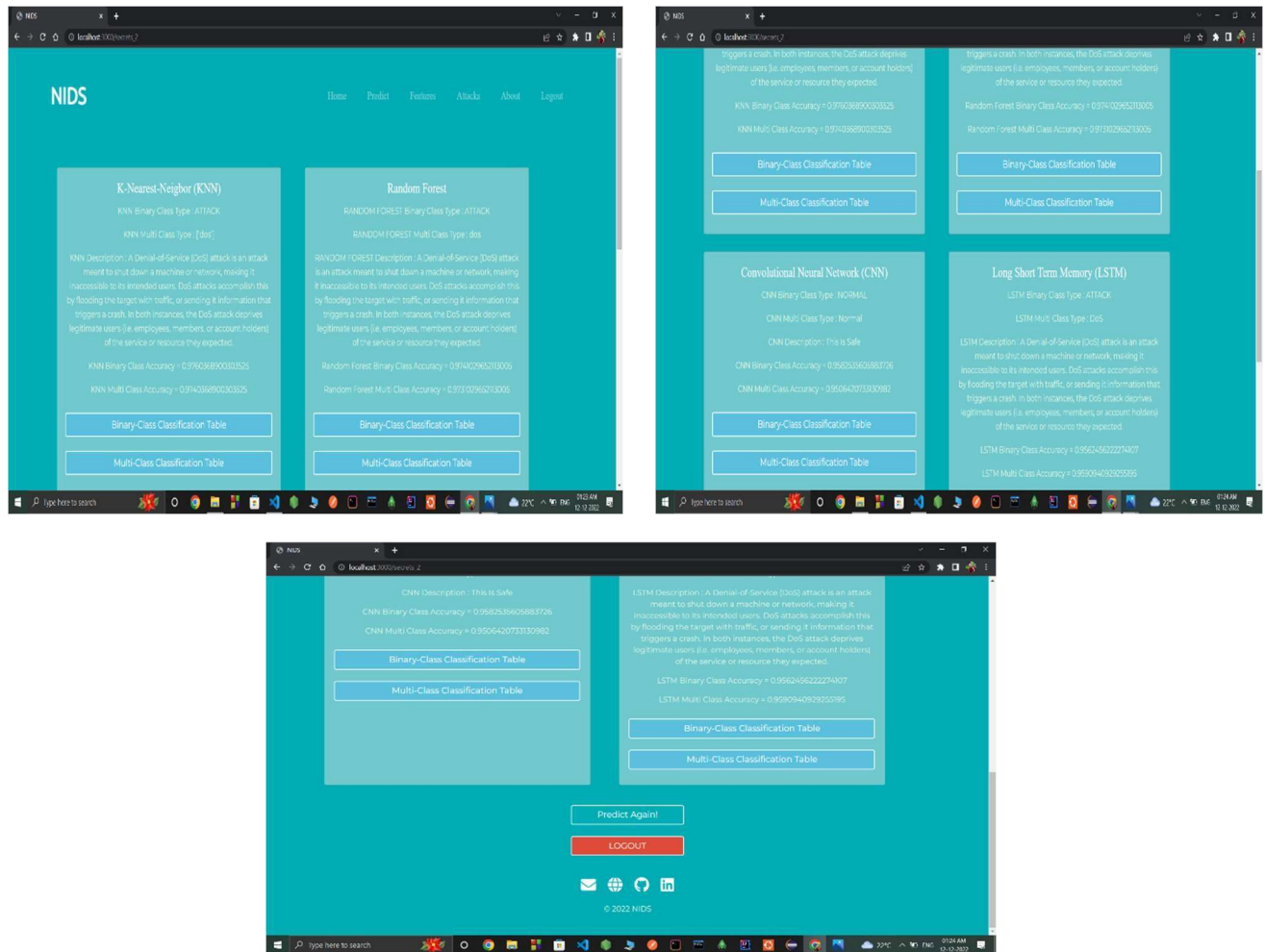
Data gets stored on MongoDB



Various options given to user to detect network



Details about each algorithm and Prediction

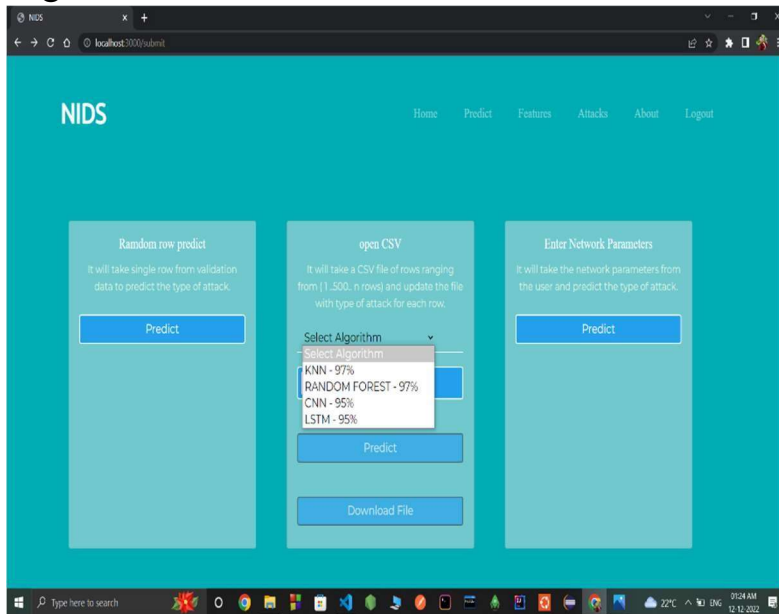


Classification table for the trained and testing data:

K-NEAREST NEIGHBOUR BINARY-CLASS CLASSIFICATION TABLE				
	PRECISION	RECALL	F1 SCORE	SUPPORT
NORMAL	0.97	0.97	0.97	9246
ATTACK	0.98	0.98	0.98	1269
ACCURACY			0.98	2145
MACRO AVERAGE	0.98	0.98	0.98	2145
WEIGHTED AVERAGE	0.98	0.98	0.98	2145

K-NEAREST NEIGHBOUR MULTI-CLASS CLASSIFICATION TABLE				
	PRECISION	RECALL	F1 SCORE	SUPPORT
DOS	0.99	0.99	0.99	6834
NORMAL	0.97	0.97	0.97	9270
PROBE	0.98	0.98	0.98	2307
U2R	0.84	0.87	0.86	919
R2L	0.96	0.98	0.97	2085
ACCURACY			0.97	2145
MACRO AVERAGE	0.95	0.96	0.97	2145

Algorithm selection:



Providing CSV as an input to model:

fs_test - Excel

FileHomeInsertPage LayoutFormulasDataReviewViewHelp

Tell me what you want to do

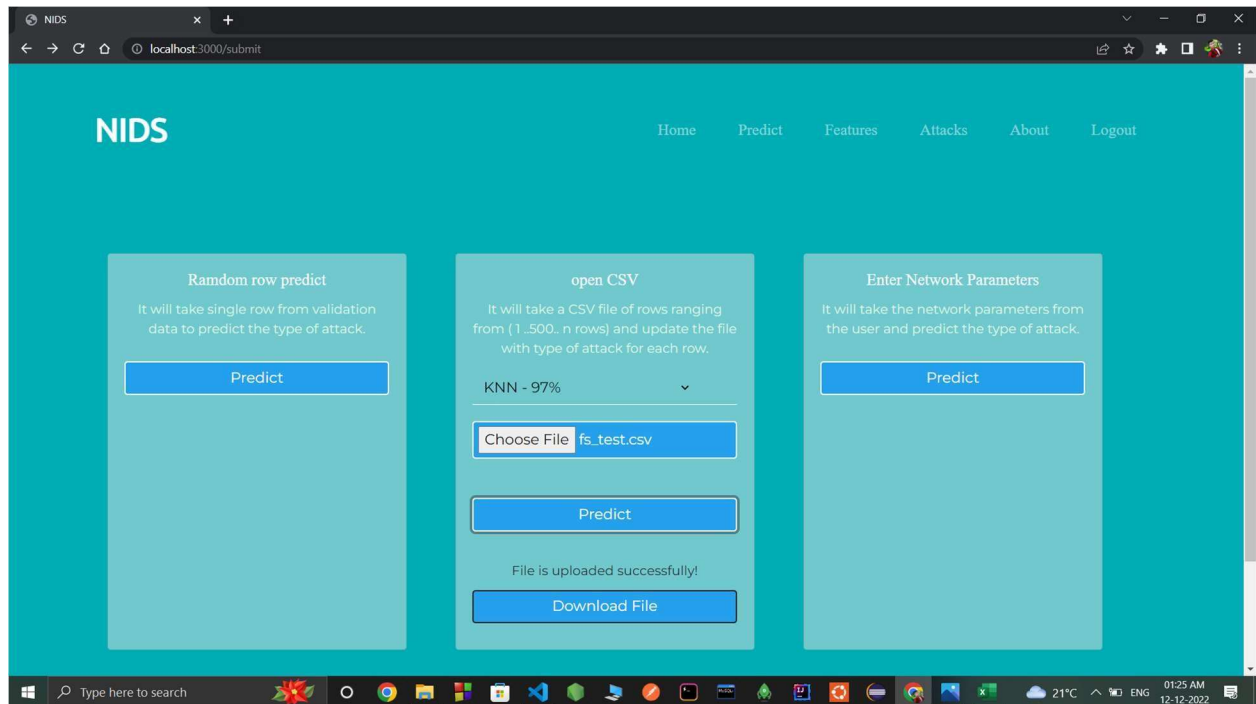
CutCopyFormat Painter

Clipboard

Calibri11

</

Prediction:



Results displayed on Excel sheet:

FileHomeInsertPage LayoutFormulasDataReviewViewHelp

CutCopyFormat PainterClipboard

Calibri11Font

Wrap Text

GeneralNumberAlignmentMerge & Center

Conditional FormattingTable

Format as Styles

Cell Styles

InsertDelete FormatCells

AutoSumFillClear

Sort & Find & FilterSelect

POSSIBLE DATA LOSS

Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

Don't show againSave As...

protocol_type

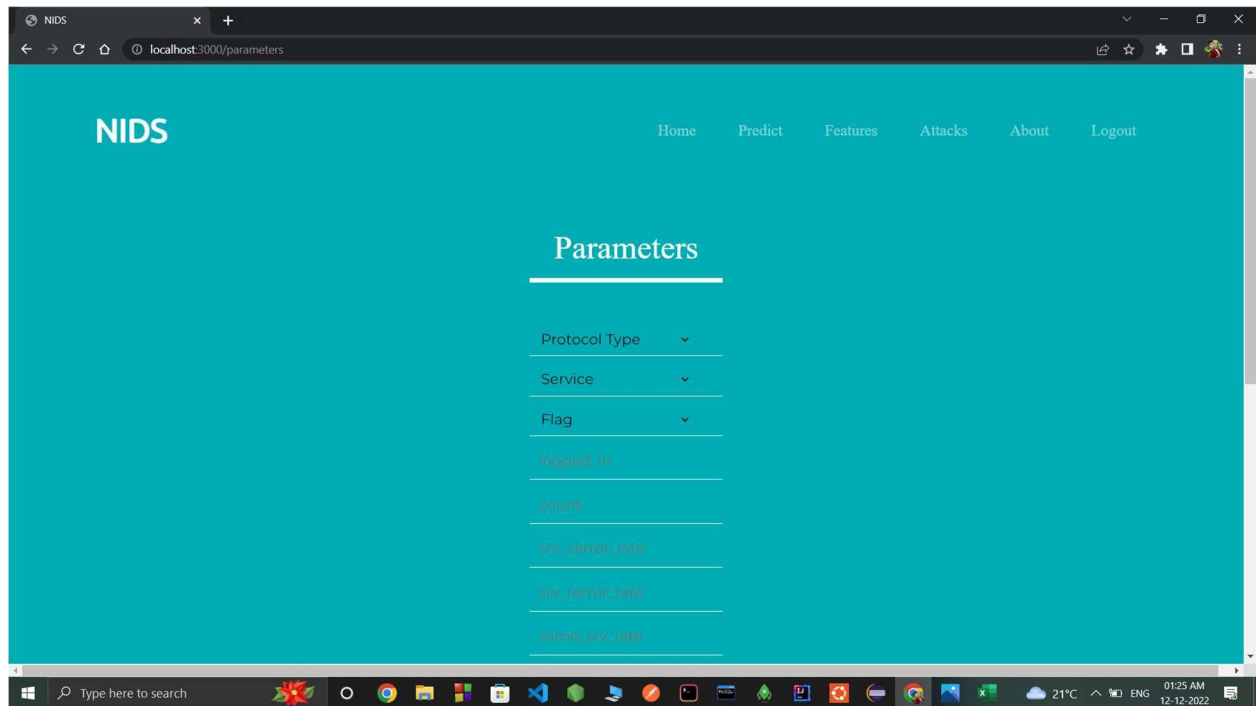
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
protocol	service	flag	logged_in	count	srv_error	srv_error	same_srv	diff_srv	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	dst_host	binary	clas	multi	class		
1	1	5	2	1	2	0.5	0	1	0	255	96	0.38	0.1	0.45	0.02	0	Normal	normal				
2	1	5	3	1	9	0	0	1	0	151	30	0.2	0.03	0.2	0	0	Normal	normal				
3	2	8	3	0	243	0	0	1	0	255	255	1	0	0.86	0	0	Normal	normal				
4	0	4	3	0	1	0	0	1	0	4	108	1	0	1	0	0	Attack	probe				
5	1	6	0	0	267	0	1	0.04	0.06	255	12	0.05	0.07	0	0	1	Attack	dos				
6	1	9	1	0	290	1	0	0.07	0.06	255	10	0.04	0.05	0	1	0	Attack	dos				
7	2	7	3	0	155	0	0	0.01	1	255	1	0	0.66	0.99	0	0	Attack	probe				
8	1	5	1	0	481	0.17	0.83	0.05	0.95	255	22	0.09	0.92	0	0.11	0.89	Attack	probe				
9	1	1	1	0	101	1	0	0.01	0.05	255	1	0	0.05	0	1	0	Attack	dos				
10	1	3	1	0	134	1	0	0.04	0.09	255	5	0.02	0.09	0	1	0	Attack	dos				
11	1	2	1	0	245	1	0	0.02	0.06	255	4	0.02	0.05	0	1	0	Attack	dos				
12	1	0	0	0	243	0	1	0.08	0.07	255	19	0.07	0.07	0	0	1	Attack	dos				
13	1	10	1	0	143	1	0	0.14	0.04	255	20	0.08	0.04	0	1	0	Attack	dos				

fs_test (34)

ReadyAccessibility: Unavailable

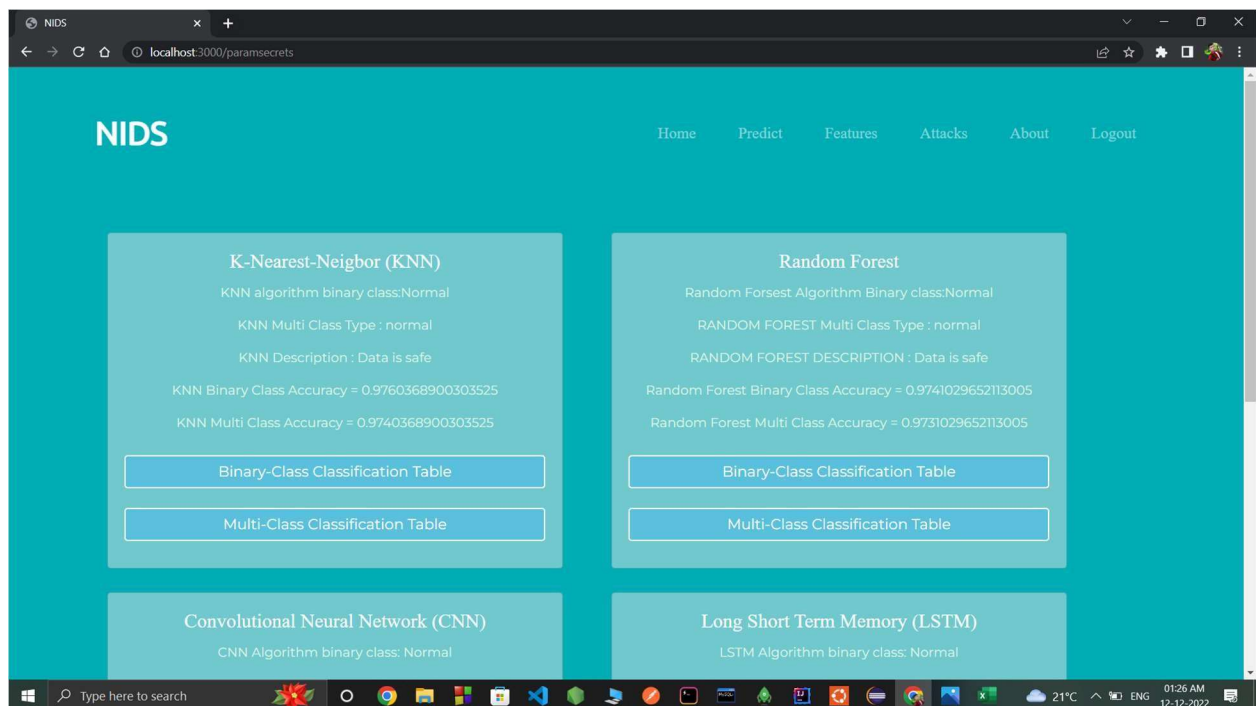
01:25 AM12-12-2022

Parameters:



The screenshot shows a web browser window with the URL `localhost:3000/parameters`. The page has a teal header with the 'NIDS' logo and navigation links: Home, Predict, Features, Attacks, About, and Logout. The main content area is titled 'Parameters' and contains several input fields: Protocol Type, Service, Flag, Logged_In, Count, src_address_ratio, src_port_ratio, and src_dst_ratio. Each field has a corresponding label and a dropdown arrow.

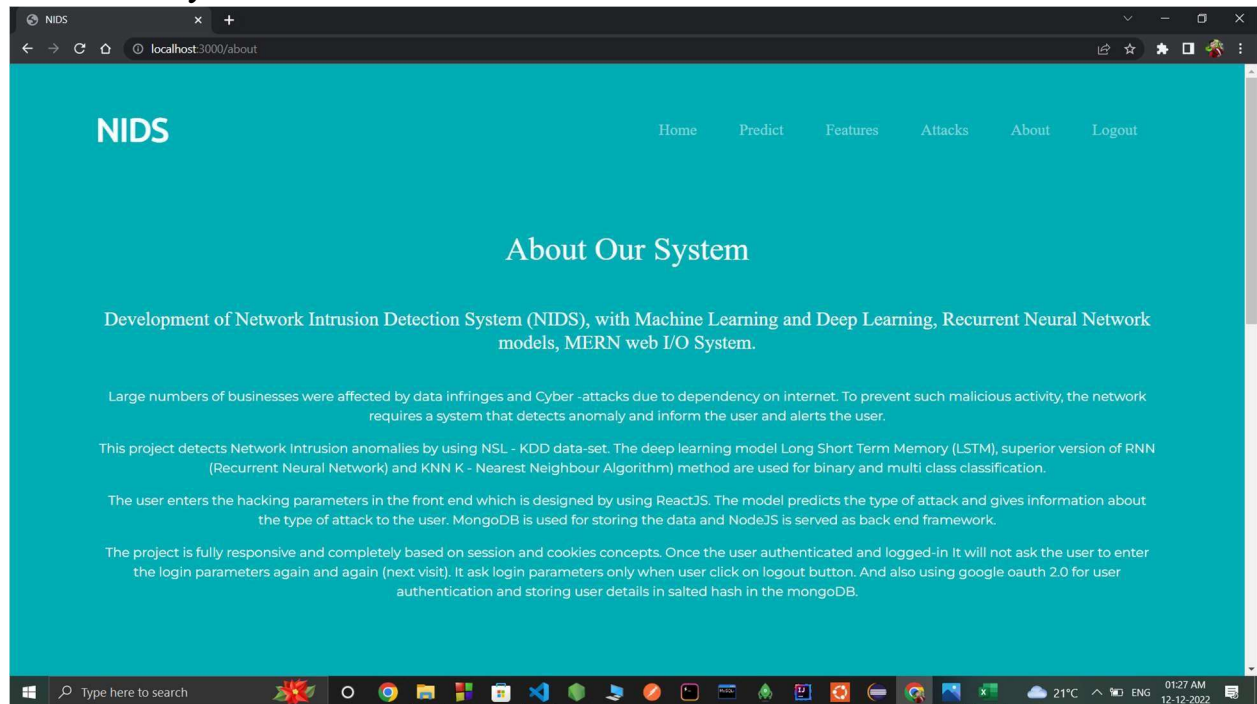
Result



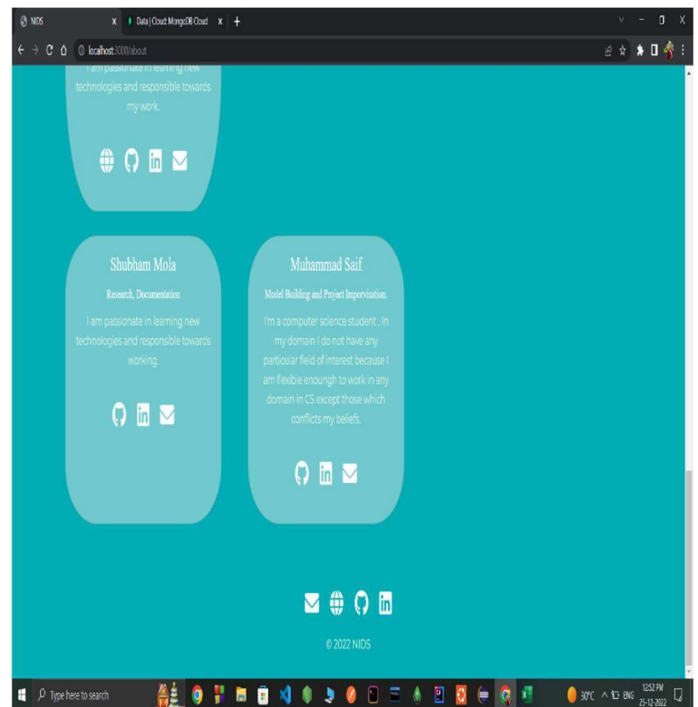
The screenshot shows the 'Results' page of the NIDS application. It features a teal header with the 'NIDS' logo and navigation links: Home, Predict, Features, Attacks, About, and Logout. The main content area is divided into four sections, each representing a different machine learning algorithm: K-Nearest-Neighbor (KNN), Random Forest, Convolutional Neural Network (CNN), and Long Short Term Memory (LSTM). Each section displays the algorithm's name, its binary class type, its multi-class type, a description, and its binary and multi-class accuracies. Below each section are two buttons: 'Binary-Class Classification Table' and 'Multi-Class Classification Table'.

Algorithm	Binary Class Type	Multi-Class Type	Description	Binary Class Accuracy	Multi-Class Accuracy
K-Nearest-Neighbor (KNN)	Normal	normal	Data is safe	0.9760368900303525	0.9740368900303525
Random Forest	Normal	normal	Data is safe	0.9741029652113005	0.9731029652113005
Convolutional Neural Network (CNN)	Normal				
Long Short Term Memory (LSTM)	Normal				

About Our System



Team Details:



Model stats:

NIDS

Home Product Features Attacks About Stats Logout

BINARY-CLASS CLASSIFICATION TABLE

	ALGORITHMS	PRECISION	RECALL	F1 SCORE	SUPPORT
NORMAL	KNN	0.97	0.97	0.97	9246
	RANDOM FOREST	0.99	0.96	0.97	9847
	CNN	0.94	0.96	0.95	9277
	LSTM	0.94	0.96	0.95	9277
ATTACK	KNN	0.98	0.98	0.98	1269
	RANDOM FOREST	0.97	0.99	0.98	11928
	CNN	0.97	0.96	0.96	12198

NIDS

Home Product Features Attacks About Stats Logout

NORMAL	KNN	0.97	0.97	0.97	9246
	RANDOM FOREST	0.99	0.96	0.97	9847
	CNN	0.94	0.96	0.95	9277
	LSTM	0.94	0.96	0.95	9277
ATTACK	KNN	0.98	0.98	0.98	1269
	RANDOM FOREST	0.97	0.99	0.98	11928
	CNN	0.97	0.96	0.96	12198
	LSTM	0.97	0.96	0.96	12198
ACCURACY	KNN			0.98	21415
	RANDOM FOREST			0.97	21415
	CNN			0.96	21415
	LSTM			0.96	21415
MACRO AVERAGE	KNN	0.98	0.98	0.98	21415
	RANDOM FOREST	0.98	0.97	0.97	21415
	CNN	0.96	0.96	0.96	21415

NIDS

Home Product Features Attacks About Stats Logout

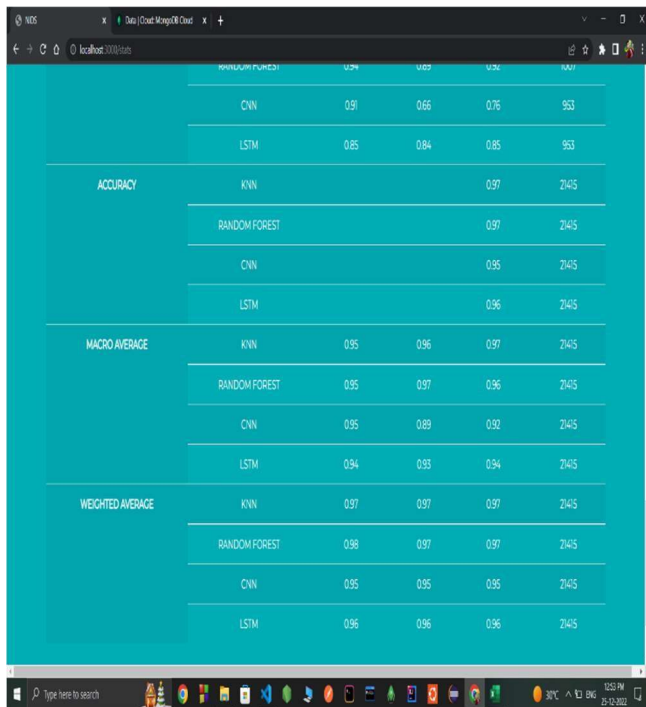
	CNN			0.96	21415
	LSTM			0.96	21415
MACRO AVERAGE	KNN	0.98	0.98	0.98	21415
	RANDOM FOREST	0.98	0.97	0.97	21415
	CNN	0.96	0.96	0.96	21415
	LSTM	0.95	0.96	0.96	21415
WEIGHTED AVERAGE	KNN	0.98	0.98	0.98	21415
	RANDOM FOREST	0.97	0.97	0.97	21415
	CNN	0.96	0.96	0.96	21415
	LSTM	0.96	0.96	0.96	21415

MULTI-CLASS CLASSIFICATION TABLE

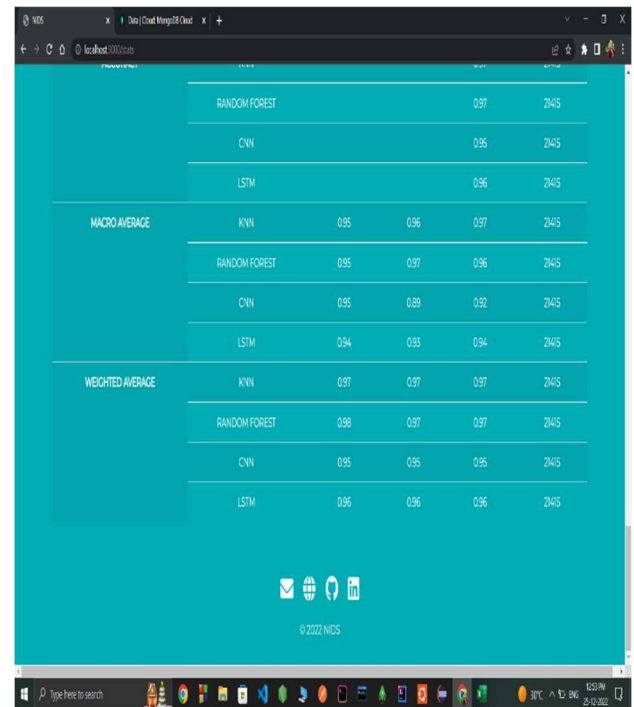
NIDS

Home Product Features Attacks About Stats Logout

	RANDOM FOREST	1.00	1.00	1.00	6622
	CNN	0.99	0.97	0.98	6622
	LSTM	1.00	0.97	0.98	6622
	KNN	0.97	0.97	0.97	9270
NORMAL	RANDOM FOREST	0.99	0.95	0.97	9913
	CNN	0.92	0.98	0.95	9277
	LSTM	0.94	0.97	0.96	9277
	KNN	0.98	0.98	0.98	2307
PROBE	RANDOM FOREST	1.00	1.00	1.00	2308
	CNN	0.94	0.97	0.96	2298
	LSTM	0.96	0.98	0.97	2298
	KNN	0.84	0.87	0.86	919
UDR	RANDOM FOREST	0.83	1.00	0.90	1765
	CNN	0.97	0.89	0.93	2125
	LSTM	0.97	0.90	0.94	2125



INDIVIDUAL MODELS		U24	U25	U26	U27
ACCURACY	CNN	0.91	0.66	0.76	953
	LSTM	0.85	0.84	0.85	953
	KNN			0.97	21415
	RANDOM FOREST			0.97	21415
MACRO AVERAGE	CNN		0.95		21415
	LSTM		0.96		21415
	KNN	0.95	0.96	0.97	21415
	RANDOM FOREST	0.95	0.97	0.96	21415
WEIGHTED AVERAGE	CNN	0.95	0.89	0.92	21415
	LSTM	0.94	0.93	0.94	21415
	KNN	0.97	0.97	0.97	21415
	RANDOM FOREST	0.98	0.97	0.97	21415



INDIVIDUAL MODELS		U24	U25	U26	U27
ACCURACY	CNN	0.91	0.66	0.76	953
	LSTM	0.85	0.84	0.85	953
	KNN			0.97	21415
	RANDOM FOREST			0.97	21415
MACRO AVERAGE	CNN		0.95		21415
	LSTM		0.96		21415
	KNN	0.95	0.96	0.97	21415
	RANDOM FOREST	0.95	0.97	0.96	21415
WEIGHTED AVERAGE	CNN	0.95	0.89	0.92	21415
	LSTM	0.94	0.93	0.94	21415
	KNN	0.97	0.97	0.97	21415
	RANDOM FOREST	0.98	0.97	0.97	21415

Future Scope:

In the future scope there can be a complete application which runs in background while browsing on to the internet and make the application raises an alert while browsing over an unsafe website.

References:

- <https://www.techopedia.com/definition/12941/network-based-intrusion-detection-system-nids>
- <https://www.ijrte.org/wp-content/uploads/papers/v9i1/A1942059120.pdf>
- <https://www.geeksforgeeks.org/k-nearest-neighbours/>
- <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>
- <https://www.unb.ca/cic/datasets/ns1.html>

Conclusion:

The project NIDS is developed considering all key aspects and primarily user data security and privacy. It is efficient and strong enough to handle any type of attack.