Chapter-4
Code

Asgi.py

"""
ASGI config for Behavior_Based_Intranet_Attacks.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see https://docs.djangoproject.com/en/3.0/howto/deployment/asgi/
""" import **os** from **django**.**core**.**asgi** import **get_asgi_application**

os.environ.**setdefault**('DJANGO_SETTINGS_MODULE',
'Behavior_Based_Intranet_Attacks.settings')

application =
get_asgi_application() **settings.py** import
os

*# Build paths inside the project like this: os.path.join(BASE_DIR,
...)*
BASE_DIR
= **os**.path.**dirname**(**os**.path.**dirname**(**os**.path.**abspath**(__file__)))

 # Quick-start development settings - unsuitable for production
 # See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
 # SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'm+1edl5m-5@u9u!b8-=4-4mq&o1%agco2xpl8c!7sn7!eowjk#'
 # SECURITY WARNING: don't run with debug turned on in production! DEBUG
= True

ALLOWED_HOSTS = []
 # Application definition
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',

```
'django.contrib.staticfiles',
  'Remote_User',
'Service_Provider',
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'Behavior_Based_Intranet_Attacks.urls'

TEMPLATES = [
{
'BACKEND':
'django.template.backends.django.DjangoTemplates',
'DIRS': [(os.path.join(BASE_DIR,'Template/htmls'))],
'APP_DIRS': True,
'OPTIONS':
{
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages
',
],
},
},
]

WSGI_APPLICATION =
'Behavior_Based_Intranet_Attacks.wsgi.application'
```

```
# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases  DATABASES =
{
'default': {
'ENGINE': 'django.db.backends.mysql',
'NAME': 'Behavior_Based_Intranet_Attacks',
'USER':'root',
'PASSWORD': 'root',
'HOST' :'127.0.0.1',
'PORT' :'3306',
}
}

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-passwordvalidators
AUTH_PASSWORD_VALIDATORS = [
{
'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityVali dator',
    }
, {
 'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
'NAME':  django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]

# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'
```

USE_I18N =
True
USE_L10N = True

USE_TZ = True

 # Static files (CSS, JavaScript, Images)
 # https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'
STATICFILES_DIRS = [**os**.path.**join**(BASE_DIR,'Template/images')]
MEDIA_URL = '/media/'
MEDIA_ROOT = **os**.path.**join**(BASE_DIR, 'Template/media')

STATIC_ROOT = '/static/'

STATIC_URL = '/static/'

Urls.py

"""graph_convolutional_networks URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/3.0/topics/http/urls/ Examples: Function views
1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns:
path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path 2.
Add a URL to urlpatterns: path('blog/', include('blog.urls')) """
from **django**.**conf**.**urls** import **url** from **django**.**contrib** import
admin from **Remote_User** import **views** as **remoteuser** from
Behavior_Based_Intranet_Attacks import **settings** from **Service_Provider** import
views as **serviceprovider** from **django**.**conf**.**urls**.**static** import **static** urlpatterns =
[**url**('admin/', **admin**.site.urls),

url(r'^$', **remoteuser**.**index**, name="index"),
url(r'^login/$', **remoteuser**.**login**, name="login"),
url(r'^Register1/$',
remoteuser.**Register1**, name="Register1"),
url(r'^Predict_Behavior_Based_Intranet_Attacks/$',
remoteuser.**Predict_Behavior_Based_Intranet_Attacks**,
name="Predict_Behavior_Based_Intranet_Attacks"),
url(r'^ViewYourProfile/$',
remoteuser.**ViewYourProfile**, name="ViewYourProfile"),
url(r'^serviceproviderlogin/$',
serviceprovider.**serviceproviderlo**
gin, name="serviceproviderlogin"),
url(r'View_Remote_Users/$',
serviceprovider.**View_Remote_Users**,nam e="View_Remote_Users"),
url(r'^charts/(?P<chart_type>\w+)', **serviceprovider**.**charts**,name="charts"),
url(r'^charts1/(?P<chart_type>\w+)', **serviceprovider**.**charts1**, name="charts1"),
url(r'^likeschart/(?P<like_chart>\w+)', **serviceprovider**.**likeschart**, name="likeschart"),
url(r'^View_Predicted_Behavior_Based_Intranet_Attacks_Type_Ratio
/$', **serviceprovider**.**View_Predicted_Behavior_Based_Intranet_Attacks_Type_ Ratio**,
name="View_Predicted_Behavior_Based_Intranet_Attacks_Type_Ratio"),
url(r'^train_model/$', **serviceprovider**.**train_model**, name="train_model"),
url(r'^View_Predicted_Behavior_Based_Intranet_Attacks_Type/$',
serviceprovider.**View_Predicted_Behavior_Based_Intranet_Attacks_Type**,
name="View_Predicted_Behavior_Based_Intranet_Attacks_Type"),
url(r'^Download_Predicted_DataSets/$', **serviceprovider**.**Download_Predicted_DataSets**,
name="Download_Predicted_DataSets"),

static(**settings**.MEDIA_URL, document_root=**settings**.MEDIA_ROOT) **Models.py** from django.db import models

 # Create your models here. from django.db.models import
CASCADE
class ClientRegister_Model(models.Model):
username = models.CharField(max_length=30)
email = models.EmailField(max_length=30)
password = models.CharField(max_length=10)
phoneno = models.CharField(max_length=10)
country = models.CharField(max_length=30)
state = models.CharField(max_length=30)
city = models.CharField(max_length=30)

```
  gender= models.CharField(max_length=30)
 address= models.CharField(max_length=30) class
Behavior_Based_Intranet_Attacks(models.Model):

Login_Frequency= models.CharField(max_length=3000)
Failed_Login_Attempts= models.CharField(max_length=3000)
Data_Transferred_MB= models.CharField(max_length=3000)
Session_Duration_Minutes= models.CharField(max_length=3000)
Accessed_Files= models.CharField(max_length=3000)
Password_Changes= models.CharField(max_length=300
  IP_Location_Changes= models.CharField(max_length=3000)
Malware_Detections= models.CharField(max_length=3000)
Suspicious_URL_Clicks= models.CharField(max_length=3000)
Time_Since_Last_Login_Hours= models.CharField(max_length=3000)
Prediction=  models.CharField(max_length=3000)
class detection_accuracy(models.Model):

names = models.CharField(max_length=300)
  ratio =  models.CharField(max_length=300)
class detection_ratio(models.Model):

names = models.CharField(max_length=300)
 ratio =  models.CharField(max_length=300)
```

userviews.py

```
from django.db.models
 import Count from django.db.models
import Q from django.shortcu
 import render, redirect, get_object_or_404
import pandas as pd  from sklearn.feature_extraction.text
import CountVectorizerfromsklearn.metrics
import accuracy_score, confusion_matrix,
classification_report
from sklearn.metrics
import accuracy_score
from sklearn.model_selection
 importtrain_test_split
  from sklearn.preprocessing
   import LabelEncoder
 from sklearn.svm import SVC
```

```
from sklearn.linear_model
 import LogisticRegression
from  sklearn.ensemble import
GradientBoostingClassifier,
VotingClassifier from sklearn.metrics
import accuracy_score
# Create your views here.
from Remote_User.models
 imporClientRegister_Model,Behavior_Based_Intranet_Attacks,detection_ratio
,detection_accuracy
def login(request): if request.method == "POST" and 'submit1' in request.POST:

username = request.POST.get('username')
  password = request.POST.get('password')
   try:  enter =
ClientRegister_Model.objects.get(username=username,password=password
)
request.session["userid"] = enter.id

return redirect('ViewYourProfile')
except: pass
return render(request,'RUser/login.html')
def index(request):
 return render(request, 'RUser/index.html')
def Add_DataSet_Details(request):
return render(request, 'RUser/Add_DataSet_Details.html',
{"excel_data":           "})
def Register1(request):  if request.method == "POST":
 username = request.POST.get('username')
 email =  request.POST.get('email')
 password = request.POST.get('password')
phoneno = request.POST.get('phoneno')
country = request.POST.get('country')
 state =  request.POST.get('state')
 city = request.POST.get('city')
   address = request.POST.get('address')
gender =   request.POST.get('gender')
ClientRegister_Model.objects.create(username=username, email=email,
password=password,  phoneno=phoneno,
 country=country,
```

```
state=state
, city=city,
address=address,
gender=gender)
obj = "Registered Successfully"
return render(request,
'RUser/Register1.html',{'object':obj})
 else:
  return render(request,'RUser/Register1.html')
def
ViewYourProfile(request):  userid  = request.session['userid']
obj  =  ClientRegister_Model.objects.get(id= userid)
   return
render(request,'RUser/ViewYourProfile.html',{
'obj ect':obj})  def
Predict_Behavior_Based_Intranet_Attacks(requ
est):     if request.method == "POST":
 if request.method == "POST":

Login_Frequency= request.POST.get('Login_Frequency')
Failed_Login_Attempts= request.POST.get('Failed_Login_Attempts')
 Data_Transferred_MB= request.POST.get('Data_Transferred_MB')
Session_Duration_Minutes= request.POST.get('Session_Duration_Minutes')
Accessed_Files= request.POST.get('Accessed_Files')
Password_Changes= request.POST.get('Password_Changes')
IP_Location_Changes= request.POST.get('IP_Location_Changes')
Malware_Detections= request.POST.get('Malware_Detections')
Suspicious_URL_Clicks= request.POST.get('Suspicious_URL_Clicks')
Time_Since_Last_Login_Hours= request.POST.get('Time_Since_Last_Login_Hours')


models = []  dataset = pd.read_csv('behavior_based_intranet_attack_data.csv')
# Preprocessing
X = dataset.drop('Attack_Type', axis=1)
   y = dataset['Attack_Type']

# Encode the target variable (Attack_Type) as it is categorical  label_encoder
= LabelEncoder() y_encoded   =
label_encoder.fit_transform(y)
```

```
# Split the data into training and testing sets
  X_train, X_test,
 y_train, y_test = train_test_split(X, y_encoded, test_size=0.3, random_state=42)

# SVM Model
  print("SVM")  from sklearn import svm

lin_clf = svm.LinearSVC()
   lin_clf.fit(X_train, y_train)
 predict_svm = lin_clf.predict(X_test)
 svm_acc =accuracy_score(y_test, predict_svm) * 100
 print("ACCURACY")
print(svm_acc)
print("CLASSIFICATION REPORT")
print(classification_report(y_test,

predict_svm))        print("CONFUSION
MATRIX") print(confusion_matrix(y_test,
predict_svm))        models.append(('svm',
lin_clf))

print("Logistic Regression")

 from sklearn.linear_model import LogisticRegression

reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train,
y_train)        y_pred = reg.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, y_pred) * 100)
 print("CLASSIFICATION
REPORT")
  print(classification_report(y_test, y_pred))
print("CONFUSION MATRIX")
 print(confusion_matrix(y_test, y_pred))
  models.append(('logistic', reg))

print("Gradient Boosting Classifier")
  gb_model = GradientBoostingClassifier()
    gb_model.fit(X_train,  y_train)
  dtcpredict = gb_model.predict(X_test)
```

```
print("ACCURACY")
print(accuracy_score(y_test,dtcpredict)*100)
 print("CLASSIFICATION REPORT")
print(classification_report(y_test,   dtcpredict))
  print("CONFUSION MATRIX")
 print(confusion_matrix(y_test, dtcpredict))
models.append(('GradientBoostingClassifier', gb_model))

classifier = VotingClassifier(models)
   classifier.fit(X_train, y_train)
   y_pred =  classifier.predict(X_test)

# Input for prediction  input_data = [[Login_Frequency, Failed_Login_Attempts,
Data_Transferred_MB,
Session_Duration_Minutes,
 Accessed_Files,
Password_Changes
, IP_Location_Changes,
 Malware_Detections,
Suspicious_URL_Clicks, Time_Since_Last_Login_Hours]]
 predicted_class_index = classifier.predict(input_data)[0]
  predicted_class = label_encoder.inverse_transform([predicted_class_index])[0]
Behavior_Based_Intranet_Attacks.objects.create(  Login_Frequency=Login_Frequency,
Failed_Login_Attempts=Failed_Login_Attempts,
Data_Transferred_MB=Data_Transferred_MB,
Session_Duration_Minutes=Session_Duration_Minutes,
Accessed_Files=Accessed_Files,
Password_Changes=Password_Changes,
IP_Location_Changes=IP_Location_Changes,
Malware_Detections=Malware_Detections,
Suspicious_URL_Clicks=Suspicious_URL_Clicks,
Time_Since_Last_Login_Hours=Time_Since_Last_Login_Hours,

Prediction=predicted_class)

return render(request,
'RUser/Predict_Behavior_Based_Intranet_Attacks.html',{'objs':predicted_class})
return render(request,
'RUser/Predict_Behavior_Based_Intranet_Attacks.html')
```

Adminviews.py

```
from django.db.models
 import  Count, Avg
 from django.shortcuts
import render, redirect
from django.db.models
import Count from django.db.models
 import Q import datetime
 import xlwt from django.http
 import HttpResponse
 import pandas as pd
from sklearn.feature_extraction.text
 import CountVectorizer
 from sklearn.metrics
import  accuracy_score,
confusion_matrix, classification_report
 from sklearn.metrics
import accuracy_score
 from sklearn.feature_extraction.text
import CountVectorizer
from sklearn.metrics
 import accuracy_score,confusion_matrix, classification_report
from sklearn.model_selection
 import train_test_split
 from sklearn.preprocessing
import LabelEncoder
from sklearn.svm
 import SVC from sklearn.linear_model
import LogisticRegression
 fromsklearn.ensemble
import     GradientBoostingClassifier,VotingClassifie     #
Create your views here.
 from Remote_User.models
  import   ClientRegister_Model,Behavior_Based_Intranet_Attacks,detection_ratio,detection _acc
uracy def serviceproviderlogin(request):
 if request.method  == "POST":
admin = request.POST.get('username')
 password = request.POST.get('password')
    if admin == "Admin" and password =="Admin":
detection_accuracy.objects.all().delete()
    return
redirect('View_Remote_Users')
```

return **render**(request,'SProvider/serviceproviderlogin.html')

def **View_Predicted_Behavior_Based_Intranet_Attacks_Type_Ratio**(request):

detection_ratio.objects.**all**().**delete**() ratio

= "" kword = 'Phishing'

print(kword)

obj = **Behavior_Based_Intranet_Attacks**.objects.**all**().**filter**(**Q**(Prediction=kword))

obj1 = **Behavior_Based_Intranet_Attacks**.objects.**all**()

count = obj.**count**(); count1 = obj1.**count**();

ratio = (count / count1) * 100 if ratio != 0: **detection_ratio**.objects.**create**(names=kword, ratio=ratio

ratio12 = "" kword12 = 'Malware'

print(kword12) obj12

Behavior_Based_Intranet_Attacks.

objects.**all**().**filter**(**Q**(Prediction=kwo

rd12)) obj112 =

Behavior_Based_Intranet_Attacks.

objects.**all**() count12 =

obj12.**count**();

count112 = obj112.**count**();

ratio12 = (count12 / count112) * 100

if ratio12 != 0:

detection_ratio.objects.**create**(names=kword12, ratio=ratio ratio13

```
kword13 = 'Brute_Force'
  print(kword13)
 obj13 =  Behavior_Based_Intranet_Attacks.objects.all().filter(Q(Prediction=kword13))
  obj113 = Behavior_Based_Intranet_Attacks.objects.all()
  count13 = obj13.count();
 count113 = obj113.count();
   ratio13 = (count13 / count113) * 100 if ratio13 != 0:
detection_ratio.objects.create(names=kword13, ratio=ratio13)
  obj = detection_ratio.objects.all() return render(request,
'SProvider/View_Predicted_Behavior_Based_Intranet_Attacks_Type_Ratio.html',
{'objs': obj})

 def View_Remote_Users(request):

obj=ClientRegister_Model.objects.all()
return
render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

 def charts(request,chart_type):

chart1 = detection_ratio.objects.values('names').annotate(dcount=Avg('ratio'))

return render(request,"SProvider/charts.html", {'form':chart1, 'chart_type':chart_type})
def              charts1(request,chart_type):
chart1  =
detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
return render(request,"SProvider/charts1.html", {'form':chart1, 'chart_type':chart_type})


 def View_Predicted_Behavior_Based_Intranet_Attacks_Type(request):

 obj =Behavior_Based_Intranet_Attacks.objects.all()

return render(request,
'SProvider/View_Predicted_Behavior_Based_Intranet_Attacks_Type.html', {'list_objects':  obj})



def likeschart(request,like_chart):

charts =detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))

 return render(request,"SProvider/likeschart.html", {'form':charts,

'like_chart':like_chart})
def     Download_Predicted_DataSets(request):
```

```
respons= HttpResponse(content_type='application/ms-excel')
# decide file name     response['Content-Disposition'] = 'attachment;
filename="Predicted_Datasets.xls"'
 # creating workbook
 wb = xlwt.Workbook(encoding='utf-8')
# adding sheet
 Ws wb.add_sheet("sheet1")
# Sheet header, first row
row_num = 0     font_style
= xlwt.XFStyle()
  # headers are bold
font_style.font.bold = True
  # writer = csv.writer(response)

    obj = Behavior_Based_Intranet_Attacks.objects.all() data = obj
    # dummy method to fetch data. for my_row in data: row_num = row_num + 1
      ws.write(row_num, 0,
      my_row.Login_Frequency, font_style)
          ws.write(row_num,
     1,my_row.Failed_Login_Attempts, font_style)
     ws.write(row_num,
    2, my_row.Failed_Login_Attempts, font_style)
  ws.write(row_num,
 3, my_row.Session_Duration_Minutes, font_style)
  ws.write(row_num,
 4, my_row.Accessed_Files, font_style)
  ws.write(row_num,
 5, my_row.Password_Changes, font_style)
ws.write(row_num,
6, my_row.IP_Location_Changes, font_style)
 ws.write(row_num,
7, my_row.Malware_Detections, font_style)
        ws.write(row_num,
8, my_row.Suspicious_URL_Clicks, font_style)
    ws.write(row_num,
9, my_row.Time_Since_Last_Login_Hours, font_style)
   ws.write(row_num,
 10, my_row.Prediction, font_style)
   wb.save(response)
 return respons
 def train_model(request):
detection_accuracy.objects.all().delete()
    models = []        dataset
```

```
pd.read_csv('behavior_based_intranet_attack_data.csv')

# Preprocessing

X = dataset.drop('Attack_Type',
axis=1)     y = dataset['Attack_Type']


# Encode the target variable (Attack_Type) as it is categorical

   label_encoder          =          LabelEncoder()

y_encoded      =

label_encoder.fit_transform(y)

    # Split the data into training and testing sets

    X_train, X_test, y_train, y_test = train_test_split(X, y_encoded,
test_size=0.3, random_state=42)
     # SVM Model     print("SVM")


    from sklearn
    import svm
  lin_clf = svm.LinearSVC()
 lin_clf.fit(X_train, y_train)
 predict_svm = lin_clf.predict(X_test)
   svm_acc = accuracy_score(y_test, predict_svm) * 100
  print(svm_acc)  print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_svm))

print("CONFUSION MATRIX")

print(confusion_matrix(y_test, predict_svm))
 models.append(('svmlin_clf))
detection_accuracy.objects.create(names="SVM",
ratio=svm_acc)     print("Logistic Regression")
  from sklearn.linear_model import LogisticRegression
   reg =LogisticRegression(random_state=0,
solver='lbfgs').fit(X_train, y_train)
y_pred = reg.predict(X_test)
 print("ACCURACY")
print(accuracy_score(y_test, y_pred) * 100)
print("CLASSIFICATION REPORT")
 print(classification_report(y_test, y_pred))
```

```
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test,y_pred))
    models.append(('logistic', reg))
     detection_accuracy.objects.create(names="Logistic Regression",
ratio=accuracy_score(y_test, y_pred) * 100)


print("GradientBoostingClassifier")     dtc =
GradientBoostingClassifier() dtc.fit(X_train, y_train)
dtcpredict = dtc.predict(X_test) print("ACCURACY")
print(accuracy_score(y_test,    dtcpredict)    *    100)
print("CLASSIFICATION REPORT")
 print(classification_report(y_test, dtcpredict))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test,dtcpredict))
models.append(('GradientBoostingClassifier',dtc))
detection_accuracy.objects.create(names="Gradient  Boosting Classifier",
 ratio=accuracy_score(y_test,  dtcpredict)  *  100)
print("VotingClassifier")
  classifier = VotingClassifier(models)
classifier.fit(X_train, y_train)
  knpredict = classifier.predict(X_test)
  print("ACCURACY")
  print(accuracy_score(y_test,knpredict) *100)
 print("CLASSIFICATION REPORT")
 print(classification_report(y_test, knpredict))
print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, knpredict))
   detection_accuracy.objects.create(names="VotingClassifier",
ratio=accuracy_score(y_test, knpredict) * 100)
    csv_format = 'Results.csv'     dataset.to_csv(csv_format, index=False)
```

```
  dataset.to_markdown      obj = detection_accuracy.objects.all()
return render(request,'SProvider/train_model.html', {'objs': obj})
```

Header.html

```
<!DOCTYPE html>

{% load static %}

<html lang="en">

<body>

<head>

<meta charset="UTF-8">

<title>Remote User</title>

<link href="https://fonts.googleapis.com/css?family=Russo+One" rel="stylesheet">

<style>       body{background: url("{% static 'bg.jpg'
%}");   background-size:cover;   fontfamily:
'Russo One', sans-serif;    backgroundcolor:
#000000;
}  h1{      color:white;

}

.topnav     {

overflow: hidden;

  backgroundcolor:

  #812;
}

.topnav a {

  float: left;

color: #FFFFFF;  text-align:

center;

padding: 14px 16px;    textdecoration: none;
font-size:   17px;

 }
```

```
.topnav a:hover
{ backgroundcolor:
#ddd; color:

black;

}

.topnav a.active
{
backgroundcolor:

#8e4fd1; color: white;

}

.style1 {color: #FF0000}

</style>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>

<body>


<h1 align="center" class="style1">An Advanced Approach for Detecting Behaviour-
Based Intranet Attacks by Machine Learning</h1>

<div class="tab-content tab-space">

  <div class="tab-pane active" id="preview-alerts">

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,500,500,200" rel="stylesheet"
/>

<script src="https://kit.fontawesome.com/42d5adcbca.js" crossorigin="anonymous"></script>

<link href="https://unpkg.com/soft-ui-design-system@1.0.1/assets/css/softdesignsystem.min.css"
rel="stylesheet" /><div class="container py-5">

<div class="alert alert-success text-white font-weight-bold" role="alert">

<a href="{% url 'Predict_Behavior_Based_Intranet_Attacks' %}">PREDICT BEHAVIOUR
BASED INTRANET ATTACKS TYPE</a>||

<a href="{% url 'ViewYourProfile' %}">VIEW YOUR PROFILE</a>||

<a href="{% url 'index' %}">LOGOUT</a>
```

```
</div>

</div>

</div>

<div class="mainholder">

{% block userblock %}

{% endblock %}

</div>

</body>

</html>
```

Index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

{% load static %}

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Home Page</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link rel="stylesheet"  type="text/css" href="{% static 'style.css'%} "  />

<link rel="stylesheet"   type="text/css" href="{% static 'coin-slider.css'%}" />

<script type="text/javascript" src="{% static 'cufon-yui.js'%} "></script>

<script type="text/javascript" src="{% static 'cufon-aller.js'%}"></script>

<script type="text/javascript" src="{% static 'jquery-1.4.2.min.js'%}"></script>

<script type="text/javascript" src="{% static 'script.js'%}"></script>

<script type="text/javascript" src="{% static 'coin-slider.min.js'%}"></script>

<style type="text/css">
```

```
<!-- .style5 {    font-size:
24px;   color:
#FF0000;

}

.style12 {font-weight: bold}

.style13 {font-size: 24px; color: #FF0000; font-weight: bold; }

.style16 {     color:    #FF0000;     font-weight:     bold;  }    -->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header_resize">

<div class="menu_nav">

<p> </p>

</div>

<div class="mainbar">

<h1 align="center"><a href="index.html"><span class="content style5">An Advanced   Approach for Detecting Behaviour-Based Intranet Attacks by Machine   Learning</span></a></h1>

<div class="tab-content tab-space">

<div class="tab-pane active" id="preview-alerts">


<link   href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"  rel="stylesheet" />

<script src="https://kit.fontawesome.com/42d5adcbca.js"  crossorigin="anonymous"></script><link  href="https://unpkg.com/softuidesignsystem@1.0.1/assets/css/soft-design-system.min.css" rel="stylesheet"

/><div class="container py-5"> <div class="row">
```

Home| Remote User | Service Provider

</div

<h2 align="center" class="style13"> An Advanced Approach for Detecting BehaviourBased Intranet Attacks by Machine Learning</h2>

An Advanced Approach for Detecting

```
BehaviourBased Intranet Attacks by Machine Learning</p>

</div>

<div class="tab-content tab-space">

<div class="tab-pane active" id="preview-alerts">


<link  href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"  rel="stylesheet"
/>

<script src="https://kit.fontawesome.com/42d5adcbca.js"  crossorigin="anonymous"></script><link
 href="https://unpkg.com/softuidesignsystem@1.0.1/assets/css/soft-design-system.min.css"
rel="stylesheet" /><div class="container py-5">

<div class="row">

<div class="alert alert-primary text-white font-weight-bold" role="alert">

<p align="center"><span class="active"><span class="style12"><a href="{% url 'login'
%}">Home|  </a><a href="{% url 'login' %}">Remote User </a>|<a href="{% url
'serviceproviderlogin' %}"> Service Provider </a></span></span></p>

</div>

<div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="sidebar">

<div class="searchform"></div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>
```

```
</div>

<div class="fbg"></div>

<div class="footer"></div>

</div>

<div align=center></div>

</body></html>
```

Login.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

{% load static %}

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Login</title>


<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />


<link rel="stylesheet"  type="text/css" href="{% static 'style.css'%} "  />

<link rel="stylesheet"   type="text/css" href="{% static 'coin-slider.css'%}" />


<script type="text/javascript" src="{% static 'cufon-yui.js'%} "></script>

<script type="text/javascript" src="{% static 'cufon-aller.js'%}"></script>

<script type="text/javascript" src="{% static 'jquery-1.4.2.min.js'%}"></script>

<script type="text/javascript" src="{% static 'script.js'%}"></script>

<script type="text/javascript" src="{% static 'coin-slider.min.js'%}"></script>

<style type="text/css">
```

```
<!-- .style5 {
font-size:
24px;   color:
#FF0000;
}

.style12 {font-weight: bold}

.style13 {font-size: 24px; color: #FF0000; font-weight: bold; }

.style16 {color: #FF0000}

.style20 {color: #FF0000; font-weight: bold; }

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header_resize">

<div class="menu_nav">

<p> </p>

</div>

<div class="mainbar">

<h1 align="center"><a href="index.html"><span class="content style5">An Advanced
Approach for Detecting Behaviour-Based Intranet Attacks by Machine  Learning</span></a></h1>



<div class="tab-content tab-space">

<div class="tab-pane active" id="preview-alerts">


<link   href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"  rel="stylesheet"
/>
```

<script src="https://kit.fontawesome.com/42d5adcbca.js" crossorigin="anonymous"></script><link href="https://unpkg.com/softuidesignsystem@1.0.1/assets/css/soft-design-system.min.css" rel="stylesheet" />

Home| Remote User | Service Provider

<h2 align="center" class="style13"> An Advanced Approach for Detecting

BehaviourBased Intranet Attacks by Machine Learning</h2>

```
<div align="center"><form method="POST" role="form">
{% csrf_token %}
<fieldset>
{% load static %}
<img src="{% static '/login.jpg' %}" alt="My image">
<p class="style5"> Login Using Your Account: </p>

<div class="form-group">
<input type="text" name="username"   placeholder="User Name" required>

<br />
<br />
</div>
<div class="form-group">
<input type="password" name="password"   placeholder="Password" required>
</div>
<div>
<p>
<input type="submit" name="submit1"  class="btn btn-md" value="Login">
<br />
</p>
</div>
<div>

<button>
```

Are You New User !!! REGISTER</button>

</fieldset>

</form>

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet" />

<script src="https://kit.fontawesome.com/42d5adcbca.js" crossorigin="anonymous"></script><link href="https://unpkg.com/softuidesignsystem@1.0.1/assets/css/soft-design-system.min.css" rel="stylesheet" />

Home| Remote User | Service Provider

```
</div>
<div class="sidebar">
<div class="searchform"></div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
  <div class="fbg"></div>
  <div class="footer"></div>
</div>
<div align=center></div>
</body>
</html>
```

Predictbehaviourattack.html

```
{% extends 'RUser/Header.html' %}
{% block userblock %}

<link rel="icon" href="images/icon.png" type="image/x-icon" />

<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Righteous" rel="stylesheet
<link href="https://fonts.googleapis.com/css?family=Fredoka+One" rel="stylesheet">
 <style>  body  {background-
color:#000000;}
.container-fluid {padding:50px;}
```

```
.container{background-color:white;padding:50px;   }

#title{font-family: 'Fredoka One', cursive;

}

.text-uppercase{      font-family:
'Righteous', cursive;  }


  .tweettext{

border: 2px solid yellowgreen;     width:
904px;     height: 202px;     overflow:
scroll;
backgroundcolor:; }
.style1 {

color: #FF0000;   fontweight: bold;
}

.style4 {color: #FFFF00; font-weight: bold; }

.style6 {
fontsize: 24px;
color:
#FFFF00;    fontweight:
bold; }

.style7 {color: #FFFF00}

.style12 {font-size: 16px}

</style>

<body>

<div class="container-fluid">

<div class="container">

<div class="row">

<div class="col-md-5">

<form role="form" method="POST" > {% csrf_token %} <fieldset>
```

<p *class*="text-uppercase pull-center style1 style12">Prediction Of <span *class*="style1">Behaviour Based Intranet Attacks !!! </p>

<hr>

{% csrf_token %}

<table *width*="1013" align="center">

<tr>

<td *height*="54" *colspan*="4" bgcolor="#FF0000"><div align="center" *class*="style6">Enter Behaviour Based Intranet Attacks Datasets Details Here !!!
</td>

</tr>
<tr>
<td *height*="44" bgcolor="#FF0000"><div align="center" *class*="style4">Enter Login_Frequency </td>

<td><input *type*="text" *name*="Login_Frequency"></td>
<td bgcolor="#FF0000"><div align="center" *class*="style4">Enter Failed_Login_Attempts </td>

<td><input *type*="text" *name*="Failed_Login_Attempts"></td>

</tr>

<tr>
<td *height*="44" bgcolor="#FF0000"><div align="center" *class*="style4">Enter Data_Transferred_MB </td>
<td><input *type*="text" *name*="Data_Transferred_MB"></td>

<td bgcolor="#FF0000"><div align="center" *class*="style4">Enter Session_Duration_Minutes </td>

<td><input *type*="text" *name*="Session_Duration_Minutes"></td>

</tr>

<tr>

<td *height*="44" bgcolor="#FF0000"><div align="center" *class*="style4">Enter Accessed_Files </td>

<td><input *type*="text" *name*="Accessed_Files"></td>
<td bgcolor="#FF0000"><div align="center" Password_Changes *class*="style4">Enter </td>

<td><input *type*="text" *name*="Password_Changes"></td>

</tr>

<tr>
<td *height*="44" bgcolor="#FF0000"><div align="center" *class*="style4">Enter IP_Location_Changes </td>

<td><input *type*="text" *name*="IP_Location_Changes"></td>

<td bgcolor="#FF0000"><div align="center" *class*="style4">Enter No Of Malware_Detections </td>

<td><input *type*="text" *name*="Malware_Detections"></td>

</tr>

<tr>
<td *height*="44" bgcolor="#FF0000"><div align="center" *class*="style4">Enter Suspicious_URL_Clicks </td>

<td><input *type*="text" *name*="Suspicious_URL_Clicks"></td>

<td bgcolor="#FF0000"><div align="center" *class*="style4">Enter No Of Time_Since_Last_Login_Hours </td>

<td><input *type*="text" *name*="Time_Since_Last_Login_Hours"></td>

</tr>

<tr>

<td *width*="217"> </td>

<td *width*="268"><input *name*="submit" *type*="submit" *class*="style1" *value*="Predict"></td>

</tr>

</table>

</fieldset>

</form>

<form *role*="form" *method*="POST" >

{% csrf_token %}

<fieldset>

<hr>

<table *width*="878" border="0" align="center" >

<tr><td *width*="614" *height*="55" bgcolor="#FF0000"> <span *class*="style6">Prediction of Health Insurance Claim Fraud Status <span *class*="style4">:: <span *class*="style7">--> </td>

<td *width*="254" bgcolor="#FFFFFF" *style*="color:red; fontsize:20px; fontfamily:fantasy" > {{objs}} </td>

</tr>

</table>

</fieldset>

</form>

<div *class*="col-md-2">

<*!*-------*null*------>

{% endblock %}

<tr>

View behaviourattacktype.html

{% extends 'SProvider/Header.html' %}

{% block researchblock %}

```
<link rel="icon" href="images/icon.png" type="image/x-icon" />

<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Righteous" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Fredoka+One" rel="stylesheet">


<style>        body

{background- color:#000000;}

.container-fluid {padding:50px;}

.container{background-color:white;padding:50px;    }

#title{font-family: 'Fredoka One', cursive;

}

.text-uppercase{

  fontfamily: 'Righteous', cursive;

}

.tweettext{

    border: 2px solid
yellowgreen;     width:  1104px;
height: 442px;     overflow:
scroll;     backgroundcolor:; }
.style1 {

color: #FF0000;      fontweight:
bold; }

.style7 {color: #FFFF00}

</style>

<body>

<div class="container-fluid">

<div class="container">  <div

class="row">
```

```
<div class="col-md-5">

<form role="form" method="POST" >

{% csrf_token %}

<fieldset>

<p align="center" class="text-uppercase pull-center style1">View
Behaviour Based Intranet Attacks Type Ratio Details</p>

<hr>

        <div>

   <table border="5" align="center" bordercolor="#FF00FF">
     <tr><td  bgcolor="#FF0000"><div  align="center"><span  class="style
                style7">Behaviour
    Based Intranet Attacks Type</span></div></td>

<td      bgcolor="#FF0000" > <div    align="center">

<span  class="style10 style7">Ratio</span></div></td>

</tr>

{% for object in objs %}

<tr>

<td  bgcolor="#FFFFFF"  style="color:red;  font-size:20px;
fontfamily:fantasy" <div align="center">
{{object.names}}</div></td>
<td bgcolor="#FF0000" style="font-family:monospace;
fontsize:19px; "><div align="center"          class="style9"><span
class="style7">{{object.ratio}}</span></div></td>

</tr>

{% endfor %}

</table>

</div>

</fieldset>

</form>
```

<div *class*="col-md-2">

<*!*-------*null*------>

{% endblock %}

<tr>

View accuracy.html

{% extends 'SProvider/Header.html' %}

{% block researchblock %}

<link *rel*="icon" *href*="images/icon.png" *type*="image/x-icon" />

<link *href*="https://fonts.googleapis.com/css?family=Lobster" *rel*="stylesheet">

<link *href*="https://fonts.googleapis.com/css?family=Righteous" *rel*="stylesheet">

<link *href*="https://fonts.googleapis.com/css?family=Fredoka+One" *rel*="stylesheet">
<style> body {background-

color:#000000;}

.container-fluid {padding:50px;}

.container{background-color:white;padding:50px; }

#title{font-family: 'Fredoka One', cursive;

}

.text-uppercase{ fontfamily:
'Righteous', cursive;

```
}

.tweettext{

border: 2px solid
yellowgreen;     width:
1104px;     height: 442px;
overflow: scroll;
background-color:; }

    .style1 {  color: #FF0000; font-weight:

bold;   }


.style12 {color: #FFFF00; font-weight: bold; }

</style>



<body>

<div class="container-fluid">

<div class="container">



<div class="row">

<div class="col-md-5">

<form role="form" method="POST" >

{% csrf_token %}

<fieldset>

<p class="text-uppercase pull-center style1">View  Prediction Of Behaviour Based Intranet Attacks Type !!! </p>

<hr>

<div class="tweettext">

<table border="5" bordercolor="#FF00FF">
```

<tr>

<td bgcolor="#FF00 *class*="style12"> *class*="style8
<d align="center"
style6">Login_Frequency </td>

<td bgcolor="#FF0000"><div align="center" <span
style6">Failed_Login_Attempts </td> *class*="style12"><span *class*="style8

<td bgcolor="#FF0000"><div align="center" *class*="style12"><span *class*="style8
style6">Data_Transferred_MB </td>

```
<td bgcolor="#FF0000"><div align="center" class="style8
class="style12"><span style6">Session_Duration_Minutes
</span></div></td>
<td bgcolor="#FF0000"><div align="center" class="style12"><span class="style8
style6">Accessed_Files</span></div></td>

<td bgcolor="#FF0000"><div align="center" class="style12"><span class="style8
style6">Password_Changes</span></div></td>
<td bgcolor="#FF0000"><div align="center" class="style12"><span class="style8
style6">IP_Location_Changes</span></div></td>
<td bgcolor="#FF0000"><div align="center" class="style12"><span class="style8
style6">Malware_Detections</span></div></td>
<td bgcolor="#FF0000"><div align="center" class="style12"><span class="style8
style6">Suspicious_URL_Clicks</span></div></td>
<td bgcolor="#FF0000"><div align="center" class="style12"><span class="style8
style6">Time_Since_Last_Login_Hours</span></div></td>
<td bgcolor="#FF0000"><div align="center" class="style12"><span class="style8
style6">Prediction</span></div></td>


</tr>

{% for object in list_objects %}

<tr>


<td bgcolor="#FFFFFF" style="color:red; font-size:20px; fontfamily:fantasy"
>{{object.Login_Frequency}}</td>
<td bgcolor="#FFFFFF" style="color:red; font-size:20px; font-family:fantasy"
>{{object.Failed_Login_Attempts}}</td>
<td bgcolor="#FFFFFF" style="color:red; font-size:20px; font-family:fantasy"
>{{object.Data_Transferred_MB}}</td>
<td bgcolor="#FFFFFF" style="color:red; font-size:20px; font-family:fantasy"
>{{object.Session_Duration_Minutes}}</td>
<td bgcolor="#FFFFFF" style="color:red; font-size:20px; font-family:fantasy"
>{{object.Accessed_Files}}</td>
```

<td bgcolor="#FFFFFF" *style*="color:red; font-size:20px; font-family:fantasy" >{{object.Accessed_Files}}</td>

<td bgcolor="#FFFFFF" *style*="color:red; font-size:20px; font-family:fantasy" >{{object.IP_Location_Changes}}</td>

<td bgcolor="#FFFFFF" *style*="color:red; font-size:20px; font-family:fantasy" >{{object.Malware_Detections}}</td>

<td bgcolor="#FFFFFF" *style*="color:red; font-size:20px; font-family:fantasy" >{{object.Suspicious_URL_Clicks}}</td>

<td bgcolor="#FFFFFF" *style*="color:red; font-size:20px; font-family:fantasy" >{{object.Time_Since_Last_Login_Hours}}</td>

<td bgcolor="#FFFFFF" *style*="color:red; font-size:20px; font-family:fantasy" >{{object.Prediction}}</td> </td>

{% endfor %}

</table>

</fieldset>

</form>

<div *class*="col-md-2">

<*!*-------*null*------>

{% endblock %}

<tr>

Accuracy graphs.html

{% extends 'SProvider/Header.html' %}

{% block researchblock %}

```
<style>  body{  background-color:  #FFFFFF;

}

.menu table{  width:100%;  textalign:center;
}

.menu table td:hover{  background:rgb(0,0,0);
}


.menu  table  td{  background:
#584b4f;
}

.menu table,.menu table th,.menu table td { border:

; border-collapse: collapse;

}

.menu table th,.menu table  td { padding:

15px; }

.topic h1{  color:white;
padding:2px;
textalign:center;
borderstyle:none;
height:100px;
width:1330px;  float:left;

}

#chartContainer{  width:

1180px;  margin-left:
250px;  margin-top: 354px;
position: fixed;  }

.sidemenu{  margin-top:
118px; }

</style>
```

```
<script > window.onload
= function() {


var chart = new CanvasJS.Chart("chartContainer",

{      animationEnabled:  true,

title:   {
text:  ""
}, data:
[{

{% if chart_type == "line" %}    type:

"line",

{% elif chart_type == "pie" %}    type:

"pie",

{% elif chart_type == "spline" %}  type:

"spline",

{% endif %} startAngle:
240,

yValueFormatString: "##0.00\"%\"",
    indexLabel: "{label} {y}", dataPoints:
[

{% for o in form %}

{y: {{o.dcount}}, label: "{{o.names}}"},

{% endfor %}

]

}] });  chart.render(); } </script>  <body>

<br>

<div class="topic">

<div class="sidemenu">
```

```
<h3 style="color:black;margin-left:60px;margin-top:80px">
<a href="{% url 'charts' 'pie' %}" >PIE CHAERT<br>
<br>
<br>
<br>
<br>
</a><a href="{% url 'charts' 'line' %}">LINE CHART</a></h3>
<h3 style="color:white;margin-left:60px;margin-top:100px;">
<a href="{% url 'charts' 'spline' %}"></a></h3>
</div>
<div id="chartContainer" class="graph"></div>
<script src="https://canvasjs.com/assets/script/canvasjs.min.js">
</script>  /div>   </body>
{% endblock %}
```