IJITCE

ISSN 2347- 3657

International Journal of
Information Technology & Computer Engineering

www.ijitce.com

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 3, 2024

EFFICIENCY-BASED DEEP ENSEMBLE FRAMEWORK FOR NETWORK ATTACK DETECTION

G LAKSHMINARAYANA[1], D NAGA RAJU[2], K LAKSHMAN KUMAR[3], M RANJITHIKUMAR REDDY[4]
[1]P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, Email: g.lakshminarayanan70@gmail.com
[2]Professor, Department of CSE, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, Email: raj2dasari@gmail.com
[3]Assistant Professor, Department of CSE, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, Email:lakshman5804@gmail.com
[4]Assistant Professor, Department of CSE, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, Email: ranjithk.reddy85@gmail.com

Abstract: Business, training, and significant distance correspondence require networks. Networks give many advantages, however security issues can think twice about classification, honesty, and protection. Network dangers including malware, hacking, and phishing are rising, causing monetary and reputational harm. The undertaking offers an AI-based robotized technique to address different security issues. This innovation recognizes and forestalls network dangers to safeguard information and arranged frameworks. The venture utilizes an ensemble model containing LSTM, RNN, and GRU DL models. These models use greater part casting a ballot to distinguish network dangers with high accuracy, safeguarding organized settings. The task adds a Voting Classifier (Random Forest + AdaBoost) and Stacking Classifier, which identifies network assaults with 100 percent accuracy.

Index terms *- Network Attack Detection, Machine Learning, Ensemble Learning, Deep Learning, Network Intrusion Detection.*

1. INTRODUCTION

Networking associates PCs to share data. Information can be shared by Ethernet, Wi-Fi, or link associations [1]. Organizing permits gadgets to share printers, document servers, and web associations. Networks have become fundamental in business, training, and day to day existence, permitting people to associate and trade data over huge distances.[33]

Organizing further develops asset sharing, correspondence, incorporated information the board, joint effort, efficiency, adaptability, and remote access [2]. Numerous dangers and security imperfections can think twice about frameworks and information with enormous systems administration applications [3]. Malware, hacking, phishing, DoS, MitM, and caricaturing are normal organization dangers. Network assaults can cause information robbery, framework interference, notoriety hurt, monetary misfortunes, reconnaissance, and foundation harm [4].

With more organization gambles, a automated attack detection system is required. AI-based frameworks might distinguish such attacks, permitting brief information robbery countermeasures [5]. These strategies assess huge volumes of organization information and distinguish risks progressively, permitting endeavors to answer quickly and effectively. ML distinguishes attacks by learning information designs. Involving such techniques in network security can help an association recognize and answer dangers [6], limiting the probability of effective assaults and safeguarding basic information and resources.[35]

Associations need network security to safeguard information protection. Thus, network security research is plentiful. A few additional connected works are referenced here. In [7], customary ML is utilized to distinguish network interruptions. The NSL-KDD dataset utilizes eleven ML calculations. Results uncover that tree-based network distinguishing proof strategies perform well. The recommended XGBoost model recognizes assaults with 97% accuracy. Similar neural network intrusion detection is finished in [8] utilizing the NSL-KDD dataset. The bidirectional LSTM procedure with a consideration component performs well in tests.

2. LITERATURE SURVEY

The Metaverse, a speculative emphasis of the Web, permits individuals to work, play, and collaborate socially in a constant web-based three dimensional virtual climate with a vivid encounter by producing practical sounds, pictures, and different sensations [1]. The Metaverse requires a completely vivid encounter, huge scope simultaneous clients, and consistent network, which presents remarkable difficulties to the 6th generation (6G) remote framework, like pervasive network, super low idleness, super high limit and unwavering quality, and severe security. Full inclusion detecting, smooth handling, reliable reserving, solid agreement, and security ought to be appropriately coordinated into the future 6G framework to give mass shoppers a vivid and bother free insight. This article shows the guide to the Metaverse regarding correspondence and systems administration in 6G [1, 2, 5], including the structure of the Metaverse, the severe necessities and difficulties for 6G to understand the Metaverse, and the basic advances to be coordinated in 6G to drive its execution, like astute detecting, digital twin (DT), space-air-ground-sea integrated network (SAGSIN), and

After 5G, scholastics and examiners expect 6G. They expect 6G [1] to drive data and public activity past 2030. Man-made intelligence will make 6G a profoundly independent shut circle network that compensates for 5G's shortcomings in correspondences, handling, and worldwide inclusion, accomplishing "AI-of things (AIoT)". In 6G [1, 2, 5], autos might be basically as significant as cell phones, and non-contaminating, safe, and completely independent vehicles will be the target [2]. Future 6G [5] vehicular knowledge should be examined to empower safe driving and traveler happiness. This paper will investigate organizing, correspondences, PCs, and knowledge, future specialized progressions and applications, and future issues and examination possibilities.

Specialists battle to distinguish unsafe web traffic to safeguard network frameworks. Aggressors can utilize network defects to acquire unapproved access through vindictive correspondences. Safeguarding against such attacks requires a powerful robotized framework that can recognize vindictive correspondences and forestall framework hurt. To recognize noxious traffic from multi-space frameworks, various robotized frameworks need to build their viability and exactness. This work utilizes ML to identify malignant interchanges precisely. [3] The method utilizes UNSW-NB15 [14] and IoTID20, which contain IoT-based and nearby organization traffic measurements, individually. The recommended method was improved to identify fake traffic from neighborhood and IoT networks with fantastic exactness by joining both datasets. Head part examination was utilized to diminish highlight build up to 30 for each dataset to consolidate them on a level plane. The proposed model purposes stacked gathering model extra boosting forest (EBF), which joins tree-based models including additional tree classifier, inclination supporting classifier, and irregular backwoods. EBF performed much better and had the most elevated exactness scores of 0.985 and 0.984 on the multi-area dataset for two and four classes, individually.[37]

Today, SDN plans face equivalent security weaknesses as regular organizations. SDN adjusts these risks [4]. A refusal of-administration attack against a concentrated regulator that directs a few switches, switches, and so on is more harming than a designated switch assault. A programmer could deal with an entire organization with a fake SDN regulator, though a switch could simply influence traffic. New security issues lie ahead for the SDN, especially for its plan. SDN security at all levels is given by three-layer engineering and programming connection points, which presents snags. Moderate SDN execution will increment security issues. [4] This work surveys the cutting edge and sorts the examination writing into a scientific categorization that stresses every proposition's essential qualities and commitments to the SDN's layers. We likewise recognize research holes that could illuminate future examination in view of existing work.

6G organizations are supposed to give moment worldwide availability and empower the change from "connected things" to "connected intelligence," where promising organization cutting can assist with guaranteeing network confirmation and administration provisioning for requesting vertical application situations. Simulated intelligence helped arrangements are liked over conventional models and calculations for 6G convoluted and dynamic cutting issues because of their huge informational indexes. Considering this [5], we present an instructional exercise on AI-assisted 6G organization cutting for network confirmation and administration provisioning to show its true capacity and the advantages of AI innovation. We propose six commonplace 6G organization cutting qualities, investigate the plausibility of simulated intelligence from various organization spaces and specialized perspectives, propose a contextual analysis on AI-assisted data transmission scaling, and present the primary difficulties and open issues for its future turn of events.

3. METHODOLOGY

i) Proposed Work:

The system detects network attacks utilizing a ensemble deep voting classifier (EDVC). This technique utilizes larger part voting to incorporate the predictions of three DL models — LSTM, RNN, and GRU — to distinguish network dangers all the more accurately and dependably [8, 29, 31]. The framework's high accuracy diminishes phony problems and missed attacks, further developing its threat detection. The innovation beats current strategies in recognizing network dangers, further developing organization security. It can rapidly and effectively assess huge organization information progressively, empowering fast assault reactions and lessening hurt or vulnerabilities.The project adds a Voting Classifier (Random Forest + AdaBoost) and Stacking Classifier, which detects network attacks with 100 percent accuracy. An easy to use Flask framework with SQLite reconciliation streamlines information exchange and signin for user testing in network protection applications. This smooth and safe association makes the system available and applicable in true organization security settings.

ii) System Architecture:

Figure 1 portrays the proposed approach's engineering. The organization assault qualities dataset is utilized for studies [9, 11, 12, 16]. Preprocessing incorporates class encoding and target assault name planning. Exploratory information examination analyzes network assault highlight patterns. For examinations, preparing and it are isolated 0.8:0.2 to test information. Typical, Dos, Remote to Neighborhood (R2L), Test, and Client to Root network assaults are distinguished utilizing the proposed technique.

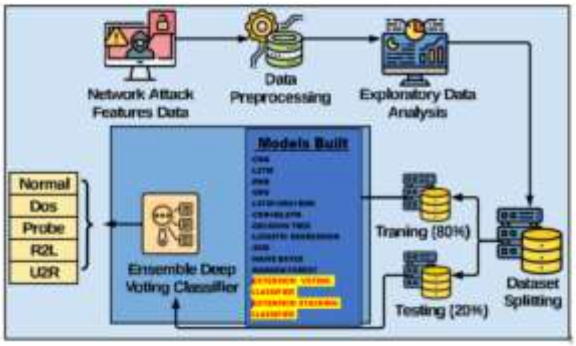
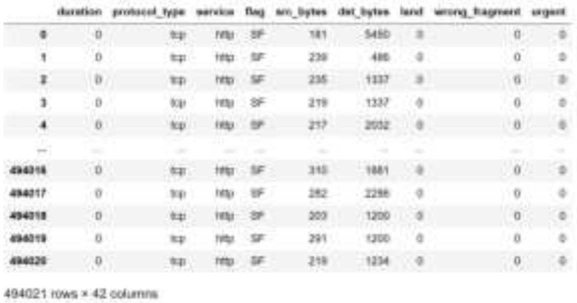


Fig 1 Proposed architecture

The task gathers information, trains complex models, and uses ensemble ways to deal with distinguish network attacks across assault classifications utilizing this engineering.

iii) Dataset collection:

The public network attack features-based NSLKDD benchmark dataset is utilized [19]. The dataset contains Dos, R2L, Test, and U2R network attacks. The assortment has 148517 records and 43 organization assault attributes. Our dataset, which incorporates all out attributes, should be preprocessed to clean and switch it over completely to mathematical information prior to taking care of it to ML models. We eliminated copies and encoded straight out attributes to preprocess the dataset. The Scikit learn LabelEncoder module encoded the 'convention type','service', and 'banner' attributes [20].



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent
0	0	tcp	http	SF	181	5450	0	0	0
1	0	tcp	http	SF	239	486	0	0	0
2	0	tcp	http	SF	235	1337	0	0	0
3	0	tcp	http	SF	219	1337	0	0	0
4	0	tcp	http	SF	217	2032	0	0	0
...	...	...	...	...	...	...	...	...	...
494016	0	tcp	http	SF	310	1881	0	0	0
494017	0	tcp	http	SF	282	2286	0	0	0
494018	0	tcp	http	SF	203	1200	0	0	0
494019	0	tcp	http	SF	291	1200	0	0	0
494020	0	tcp	http	SF	219	1234	0	0	0

494021 rows × 42 columns

Fig 2 KDD dataset

iv) Data Processing:

Data processing transforms crude information into business-helpful data. Information researchers assemble, put together, clean, check, examine, and orchestrate information into charts or papers. Information can be handled physically, precisely, or electronically. Data ought to be more important and decision-production simpler. Organizations might improve tasks and settle on basic decisions quicker.

PC programming advancement and other computerized data processing innovations add to this. Enormous information can be transformed into pertinent bits of knowledge for quality administration and independent direction.[39]

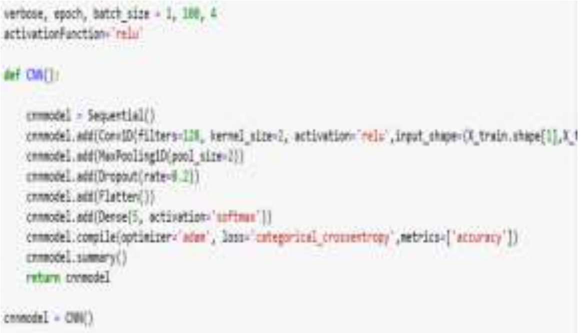
v) Feature selection:

Feature selection chooses the steadiest, non-repetitive, and pertinent elements for model turn of events. As data sets extend in amount and assortment, purposefully bringing down their size is significant. The fundamental reason for feature selection is to increment prescient model execution and limit processing cost.

One of the vital pieces of feature engineering is picking the main attributes for machine learning algorithms. To diminish input factors, feature selection methodologies take out copy or superfluous elements and limit the assortment to those generally critical to the ML model. Rather than permitting the ML model pick the main qualities, feature selection ahead of time enjoys a few benefits.

vi) Algorithms:

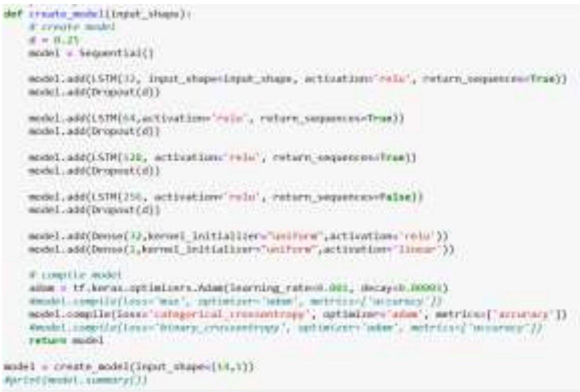
CNN (Convolutional Neural Network) - The essential utilization of CNN, a deep learning model, is picture investigation. It could be changed for this reason to remove features from network information and spot patterns or irregularities in the geological association of the information [20].



```
verbose, epoch, batch_size = 1, 100, 4
activationFunction='relu'

def CNN():

    cnnmodel = Sequential()
    cnnmodel.add(Conv1D(filters=128, kernel_size=2, activation='relu',input_shape=(X_train.shape[1],X_t
    cnnmodel.add(MaxPooling1D(pool_size=2))
    cnnmodel.add(Dropout(rate=0.2))
    cnnmodel.add(Flatten())
    cnnmodel.add(Dense(5, activation='softmax'))
    cnnmodel.compile(optimizer='adam', loss='categorical_crossentropy',metrics=['accuracy'])
    cnnmodel.summary()
    return cnnmodel

cnnmodel = CNN()
```

Fig 3 CNN

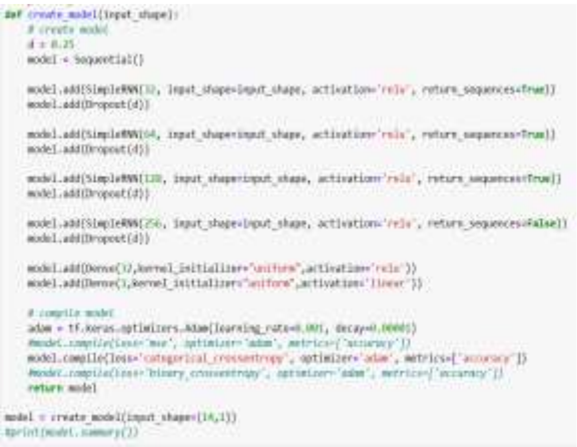
LSTM (Long Short-Term Memory) - Recurrent neural networks (RNNs) of the LSTM type are appropriate to handling successive information. It is valuable for distinguishing assault ways of behaving that change over the long haul since it tends to be utilized to the catch of examples in time series network information [29].



```
def create_model(input_shape):
    # create model
    d = 0.25
    model = Sequential()

    model.add(LSTM(32, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(64,activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(128, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(256, activation='relu', return_sequences=False))
    model.add(Dropout(d))

    model.add(Dense(32,kernel_initializer="uniform",activation='relu'))
    model.add(Dense(1,kernel_initializer="uniform",activation='linear'))

    # compile model
    adam = tf.keras.optimizers.Adam(learning_rate=0.001, decay=0.00001)
    #model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    #model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model = create_model(input_shape=(14,1))
#print(model.summary())
```

Fig 4 LSTM

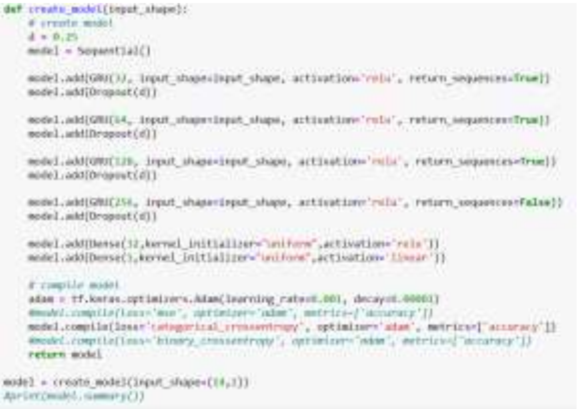
RNN (Recurrent Neural Network) - Another deep learning model for successive data processing is the RNN. It aids the recognizable proof of organization attacks with transient elements by distinguishing examples and connections in network information arrangements [27].



```
def create_model(input_shape):
    # create model
    d = 0.25
    model = Sequential()

    model.add(SimpleRNN(32, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(SimpleRNN(64, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(SimpleRNN(128, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(SimpleRNN(256, input_shape=input_shape, activation='relu', return_sequences=False))
    model.add(Dropout(d))

    model.add(Dense(32,kernel_initializer="uniform",activation='relu'))
    model.add(Dense(1,kernel_initializer="uniform",activation='linear'))

    # compile model
    adam = tf.keras.optimizers.Adam(learning_rate=0.001, decay=0.00001)
    #model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    #model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model = create_model(input_shape=(14,1))
#print(model.summary())
```

Fig 5 RNN

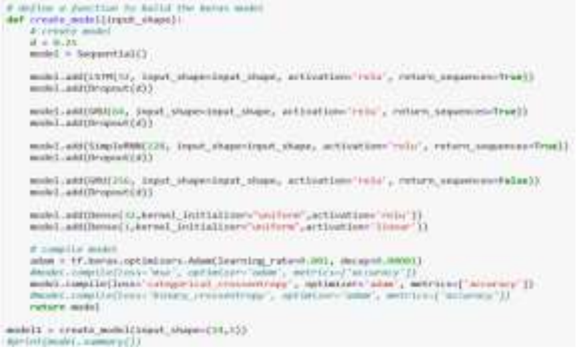
GRU (Gated Recurrent Unit) - A form on RNN that keeps the capacity to record successive examples yet works on the engineering is called GRU. It is valuable for displaying conditions in network information and can assist with recognizing assaults that incorporate worldly parts [28].



```
def create_model(input_shape):
    # create model
    d = 0.25
    model = Sequential()

    model.add(GRU(32, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(GRU(64, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(GRU(128, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(GRU(256, input_shape=input_shape, activation='relu', return_sequences=False))
    model.add(Dropout(d))

    model.add(Dense(32,kernel_initializer="uniform",activation='relu'))
    model.add(Dense(1,kernel_initializer="uniform",activation='linear'))

    # compile model
    adam = tf.keras.optimizers.Adam(learning_rate=0.001, decay=0.00001)
    #model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    #model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model = create_model(input_shape=(14,1))
#print(model.summary())
```

Fig 6 GRU

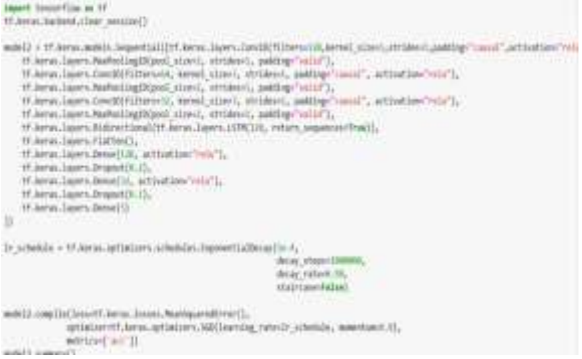
LSTM+GRU+RNN - By consolidating the benefits of GRU, RNN, and LSTM, this ensemble gives a more dependable technique to identifying network attacks. It can effectively catch different examples in network information by using different recurrent neural network architectures.



```
# define a function to build the keras model
def create_model(input_shape):
    # create model
    d = 0.25
    model = Sequential()

    model.add(LSTM(32, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(GRU(64, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(SimpleRNN(128, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(GRU(256, input_shape=input_shape, activation='relu', return_sequences=False))
    model.add(Dropout(d))

    model.add(Dense(32,kernel_initializer="uniform",activation='relu'))
    model.add(Dense(1,kernel_initializer="uniform",activation='linear'))

    # compile model
    adam = tf.keras.optimizers.Adam(learning_rate=0.001, decay=0.00001)
    #model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    #model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model1 = create_model(input_shape=(14,1))
#print(model1.summary())
```

Fig 7 LSTM+GRU+RNN

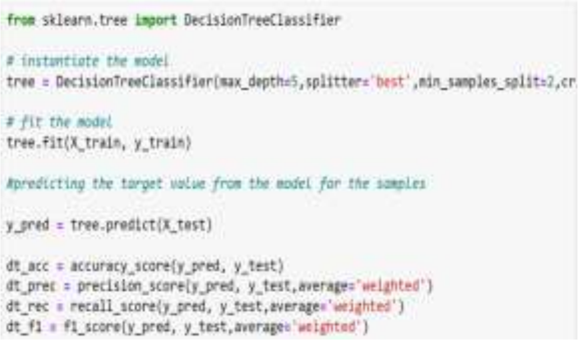
CNN+BiLSTM (Bidirectional LSTM) - Feature extraction and sequence analysis are made conceivable by the blend of CNN and BiLSTM. It improves the ability to perceive refined attacks by catching transient and geological examples in network information.



```
import tensorflow as tf
tf.keras.backend.clear_session()

model2 = tf.keras.models.Sequential([tf.keras.layers.Conv1D(filters=128,kernel_size=1,strides=1,padding="causal",activation="relu
    tf.keras.layers.MaxPooling1D(pool_size=1, strides=1, padding="valid"),
    tf.keras.layers.Conv1D(filters=64, kernel_size=1, strides=1, padding="causal", activation="relu"),
    tf.keras.layers.MaxPooling1D(pool_size=1, strides=1, padding="valid"),
    tf.keras.layers.Conv1D(filters=32, kernel_size=1, strides=1, padding="causal", activation="relu"),
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=2, padding="valid"),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128, return_sequences=True)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(16, activation="relu"),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(1)
])

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(1e-4,
                                                             decay_steps=100000,
                                                             decay_rate=0.96,
                                                             staircase=False)

model2.compile(loss=tf.keras.losses.MeanSquaredError(),
              optimizer=tf.keras.optimizers.SGD(learning_rate=lr_schedule, momentum=0.9),
              metrics=['acc'])
model2.summary()
```

Fig 8 CNN + BiLSTM

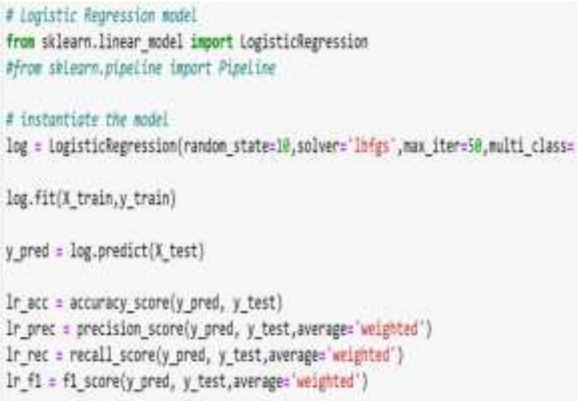
Decision Tree - Among the techniques for supervised learning are decision trees. They might be applied to the undertaking to foster a dynamic model that utilizes qualities and their associations with distinguish network dangers.



```
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth=5,splitter='best',min_samples_split=2,cr

# fit the model
tree.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = tree.predict(X_test)

dt_acc = accuracy_score(y_pred, y_test)
dt_prec = precision_score(y_pred, y_test,average='weighted')
dt_rec = recall_score(y_pred, y_test,average='weighted')
dt_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 9 decision tree

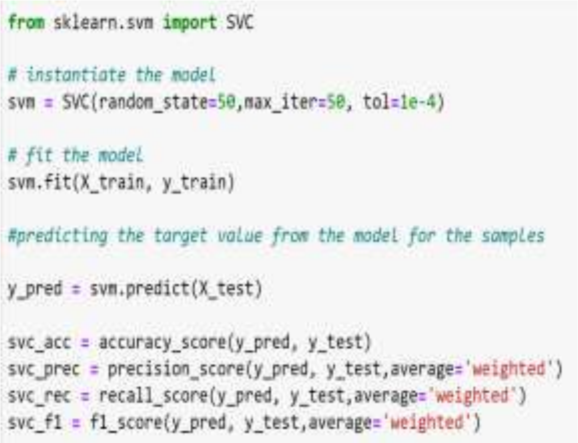
Logistic Regression - A typical factual model for double characterization is strategic relapse. It very well may be valuable in this present circumstance to sort network information as an occasion of an assault or a regular one [22].



```
# Logistic Regression model
from sklearn.linear_model import LogisticRegression
#from sklearn.pipeline import Pipeline

# instantiate the model
log = LogisticRegression(random_state=10,solver='lbfgs',max_iter=50,multi_class=

log.fit(X_train,y_train)

y_pred = log.predict(X_test)

lr_acc = accuracy_score(y_pred, y_test)
lr_prec = precision_score(y_pred, y_test,average='weighted')
lr_rec = recall_score(y_pred, y_test,average='weighted')
lr_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 10 Logistic regression

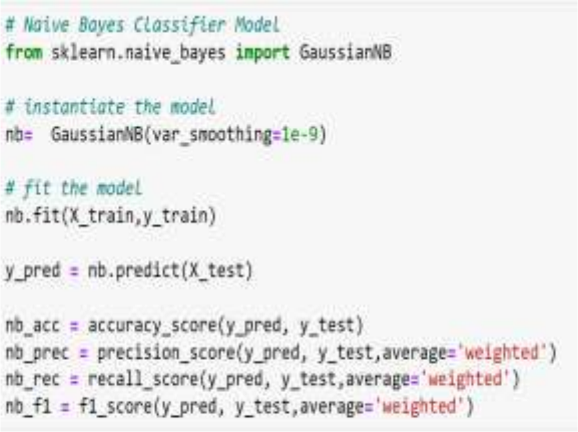
SVM (Support Vector Machine) - An ML technique called SVM isolates information into numerous orders. It is valuable for separating between genuine organization traffic and organization attacks [23].



```
from sklearn.svm import SVC

# instantiate the model
svm = SVC(random_state=50,max_iter=50, tol=1e-4)

# fit the model
svm.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = svm.predict(X_test)

svc_acc = accuracy_score(y_pred, y_test)
svc_prec = precision_score(y_pred, y_test,average='weighted')
svc_rec = recall_score(y_pred, y_test,average='weighted')
svc_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 11 SVM

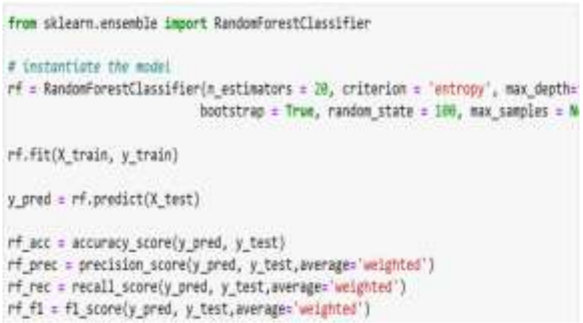
Naïve Bayes - A probabilistic calculation for order undertakings is called Naïve Bayes. It can aid classification by ascertaining the probability that network information will can be categorized as one of two sorts (assault or typical).[40]



```
# Naive Bayes Classifier Model
from sklearn.naive_bayes import GaussianNB

# instantiate the model
nb=  GaussianNB(var_smoothing=1e-9)

# fit the model
nb.fit(X_train,y_train)

y_pred = nb.predict(X_test)

nb_acc = accuracy_score(y_pred, y_test)
nb_prec = precision_score(y_pred, y_test,average='weighted')
nb_rec = recall_score(y_pred, y_test,average='weighted')
nb_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 12 Naïve bayes

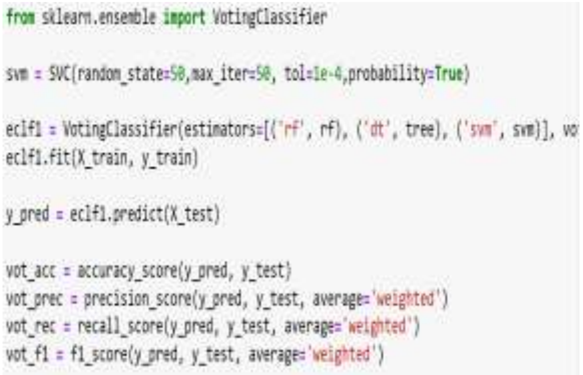
Random Forest - A few decision trees are joined in Random Forest, an ensemble learning procedure, to further develop classification accuracy. It could be applied to fabricate a stronger model for spotting network intrusions [25, 26].



```
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
rf = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', max_depth=
                            bootstrap = True, random_state = 100, max_samples = N

rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

rf_acc = accuracy_score(y_pred, y_test)
rf_prec = precision_score(y_pred, y_test,average='weighted')
rf_rec = recall_score(y_pred, y_test,average='weighted')
rf_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 13 Random forest

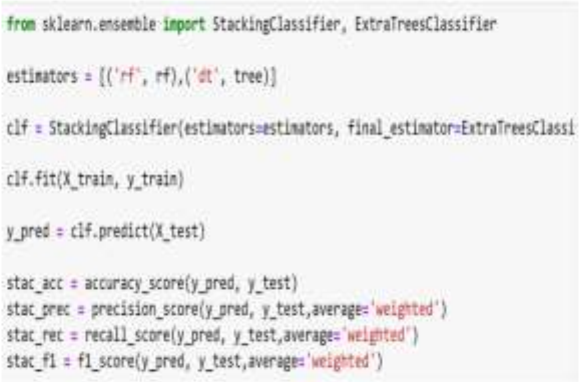
Voting Classifier - Expectations from many ML models are joined in the voting classifier. Here, consolidating data from a few models, expanding the accuracy of attack detection overall is utilized.



```
from sklearn.ensemble import VotingClassifier

svm = SVC(random_state=50,max_iter=50, tol=1e-4,probability=True)

eclf1 = VotingClassifier(estimators=[('rf', rf), ('dt', tree), ('svm', svm)], vo
eclf1.fit(X_train, y_train)

y_pred = eclf1.predict(X_test)

vot_acc = accuracy_score(y_pred, y_test)
vot_prec = precision_score(y_pred, y_test, average='weighted')
vot_rec = recall_score(y_pred, y_test, average='weighted')
vot_f1 = f1_score(y_pred, y_test, average='weighted')
```

Fig 14 Voting classifier

Stacking Classifier- Stacking utilizes a meta-classifier to total the predictions from a few models. By representing various model results and offering a more exact end, it further develops the network attack detection system.



```
from sklearn.ensemble import StackingClassifier, ExtraTreesClassifier

estimators = [('rf', rf),('dt', tree)]

clf = StackingClassifier(estimators=estimators, final_estimator=ExtraTreesClassi

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

stac_acc = accuracy_score(y_pred, y_test)
stac_prec = precision_score(y_pred, y_test,average='weighted')
stac_rec = recall_score(y_pred, y_test,average='weighted')
stac_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 15 Stacking classifier

4. EXPERIMENTAL RESULTS

Precision: Precision estimates the level of positive cases or tests precisely sorted. Precision is determined utilizing the recipe:

$$\text{Precision} = \text{True positives}/\ (\text{True positives} + \text{False positives}) = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

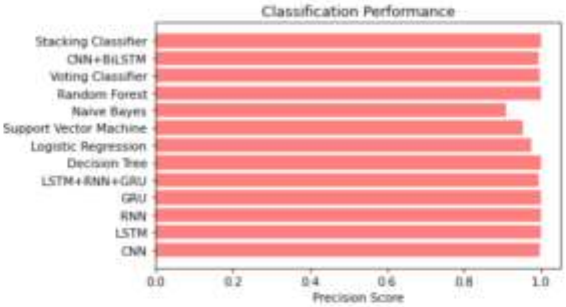


Fig 16 Precision comparison graph

Recall: Machine learning recall assesses a model's ability to perceive all significant examples of a class. It shows a model's culmination in catching occasions of a class by contrasting accurately anticipated positive perceptions with complete positives.

$$Recall = \frac{TP}{TP + FN}$$

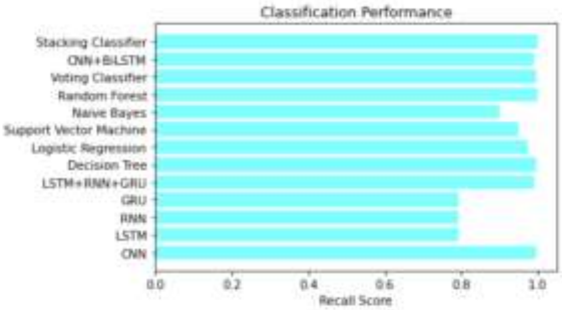


Fig 17 Recall comparison graph

Accuracy: A test's accuracy is its ability to recognize debilitated from sound cases. To quantify test accuracy, figure the small part of true positive and true negative in completely broke down cases. Numerically, this is:

$$\text{Accuracy} = \text{TP} + \text{TN}\ \text{TP} + \text{TN} + \text{FP} + \text{FN}.$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

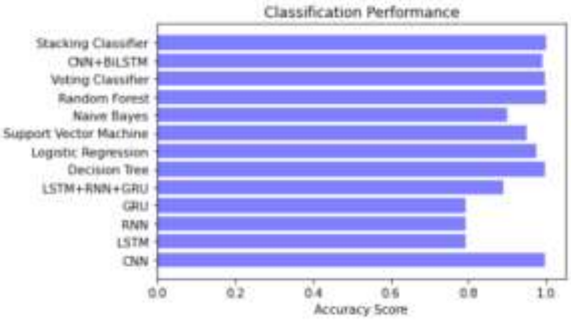


Fig 18 Accuracy graph

F1 Score: Machine learning model accuracy is estimated by F1 score. Consolidating model precision and recall scores. The accuracy measurement estimates how frequently a model anticipated accurately all through the dataset.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

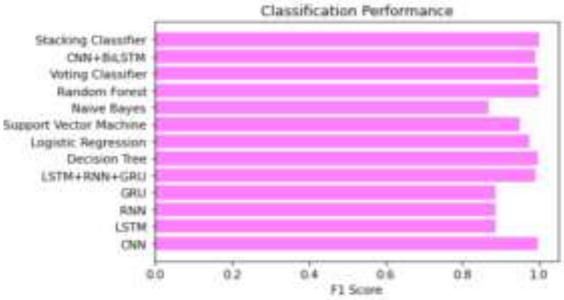
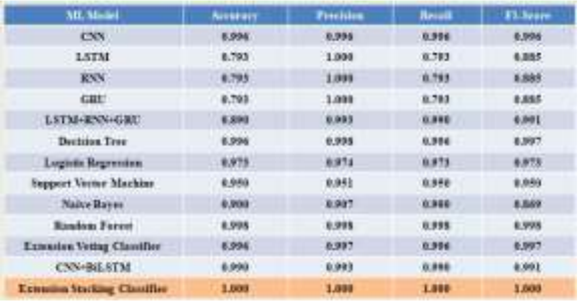


Fig 19 F1Score



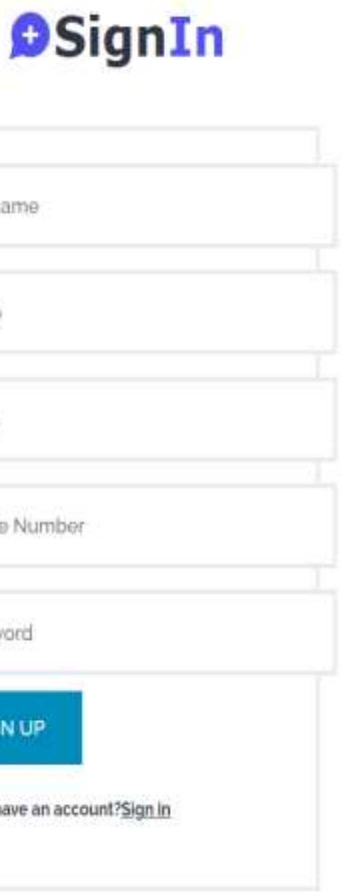
ML Model	Accuracy	Precision	Recall	F1-Score
CNN	0.996	0.996	0.996	0.996
LSTM	0.793	1.000	0.793	0.885
RNN	0.793	1.000	0.793	0.885
GRU	0.793	1.000	0.793	0.885
LSTM+RNN+GRU	0.890	0.993	0.890	0.901
Decision Tree	0.996	0.998	0.996	0.997
Logistic Regression	0.973	0.974	0.973	0.973
Support Vector Machine	0.950	0.951	0.950	0.950
Naive Bayes	0.900	0.907	0.900	0.869
Random Forest	0.998	0.998	0.998	0.998
Extension Voting Classifier	0.996	0.997	0.996	0.997
CNN+BiLSTM	0.990	0.993	0.990	0.991
Extension Stacking Classifier	1.000	1.000	1.000	1.000

Fig 20 Performance Evaluation

Welcome to Dashboard

Deep Ensemble-based Efficient Framework for Network Attack Detection

Fig 21 Home page



SignIn

Username

Name

Email

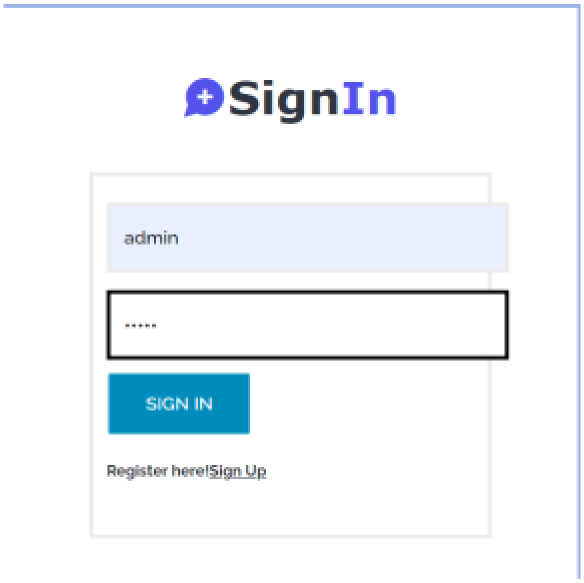
Mobile Number

Password

SIGN UP

Already have an account?Sign In

Fig 22 Signin page



SignIn

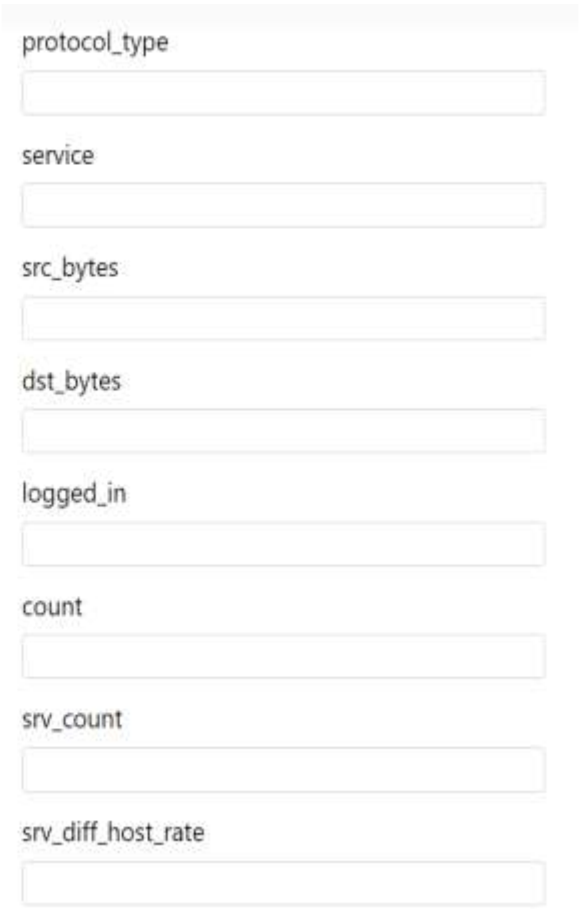
admin

······

SIGN IN

Register here!Sign Up

Fig 23 Login page



protocol_type

service

src_bytes

dst_bytes

logged_in

count

srv_count

srv_diff_host_rate

Fig 24 User input

Result: Attack is Detected and its R2L Attack!

Fig 25 Predict result for given input

5. CONCLUSION

To recognize network attacks, the venture utilizes strong DL models including LSTM, RNN, and GRU [27, 29]. The venture means to further develop PC network security by utilizing these models to safeguard against various organization dangers. The NSL-KDD dataset approves the venture's system's network attack detection accuracy. This exact approval shows the system's common sense and ability to perceive and answer PC network malware. DL models and outfit approaches empower the structure to adjust to changing assault designs. This flexibility assists the organization with opposing new assaults and keeps the structure effective in tending to network protection's dynamic nature. Ensemble approaches like Voting Classifier and Stacking Classifier further develop network attack detection in the undertaking. System testing is improved with an easy to use Flask interface and secure verification. This connection point makes execution assessment information passage simple for network protection trained professionals. Ensemble techniques and easy to use highlights further develop the system's genuine presentation.

6. FUTURE SCOPE

Future examination will analyze profound learning model plan to further develop framework processing effectiveness. This upgrade will further develop network threat detection, making the framework strong and practical. The attention on framework versatility will make it more suitable for true organization security arrangements. This improvement will limit functional costs and permit the framework to oversee bigger and more muddled networks. Advancement of continuous attack detection techniques is arranged [19]. This ongoing limit will permit the framework to adjust and kill assaults, diminishing damage and powerlessness rapidly. The drive will change the structure to oblige huge scope network information as amounts extend. The framework should adjust to the developing intricacy and volume of organization traffic to remain compelling. Future improvement should add peculiarity recognition in the system. This upgrade permits the framework to find new assault designs and respond to new dangers, empowering proactive network protection.[42]

REFERENCES

[1] F. Tang, X. Chen, M. Zhao, and N. Kato, “The roadmap of communication and networking in 6g for the metaverse,” IEEE Wireless Communications, 2022.

[2] H. Guo, X. Zhou, J. Liu, and Y. Zhang, “Vehicular intelligence in 6g: Networking, communications, and computing,” Vehicular Communications, vol. 33, p. 100399, 2022.

[3] P. L. Indrasiri, E. Lee, V. Rupapara, F. Rustam, and I. Ashraf, “Malicious traffic detection in iot and local networks using stacked ensemble classifier,” Computers, Materials and Continua, vol. 71, no. 1, pp. 489– 515, 2022.

[4] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on sdn security: threats, mitigations, and future directions," Journal of Reliable Intelligent Environments, pp. 1–39, 2022.

[5] J. Wang, J. Liu, J. Li, and N. Kato, "Artificial intelligence-assisted network slicing: Network assurance and service provisioning in 6g," IEEE Vehicular Technology Magazine, 2023.

[6] M. A. Talukder, K. F. Hasan, M. M. Islam, M. A. Uddin, A. Akhter, M. A. Yousuf, F. Alharbi, and M. A. Moni, "A dependable hybrid machine learning model for network intrusion detection," Journal of Information Security and Applications, vol. 72, p. 103405, 2023.

[7] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for contiki-ng-based iot networks exposed to nsl-kdd dataset," in Proceedings of the 2nd ACM workshop on wireless security and machine learning, 2020, pp. 25–30.

[8] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "Bat: Deep learning methods on network intrusion detection using nsl-kdd dataset," IEEE Access, vol. 8, pp. 29 575–29 585, 2020.

[9] G. C. Amaizu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Investigating network intrusion detection datasets using machine learning," in 2020 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2020, pp. 1325–1328.

[10] M. Esmaeili, S. H. Goki, B. H. K. Masjidi, M. Sameh, H. Gharagozlou, and A. S. Mohammed, "Ml-ddosnet: Iot intrusion detection based on denial-of-service attacks using machine learning methods and nsl-kdd," Wireless Communications and Mobile Computing, vol. 2022, 2022.

[11] A. K. Balyan, S. Ahuja, U. K. Lilhore, S. K. Sharma, P. Manoharan, A. D. Algarni, H. Elmannai, and K. Raahemifar, "A hybrid intrusion detection model using ega-pso and improved random forest method," Sensors, vol. 22, no. 16, p. 5986, 2022.

[12] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," IEEE access, vol. 8, pp. 32 464–32 476, 2020.

[13] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning," Ieee Access, vol. 9, pp. 75 729–75 740, 2021.

[14] S. Cherfi, A. Boulaiche, and A. Lemouari, "Multi-layer perceptron for intrusion detection using simulated annealing," in Modelling and Implementation of Complex Systems: Proceedings of the 7th International Symposium, MISC 2022, Mostaganem, Algeria, October 30-31, 2022. Springer, 2022, pp. 31–45.

[15] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," Future Internet, vol. 13, no. 5, p. 111, 2021.

[16] T. Wisanwanichthan and M. Thammawichai, "A double-layered hybrid approach for network intrusion detection system using combined naive bayes and

svm,” IEEE Access, vol. 9, pp. 138 432–138 450, 2021.

[17] N. Sahar, R. Mishra, and S. Kalam, “Deep learning approach-based network intrusion detection system for fog-assisted iot,” in Proceedings of international conference on big data, machine learning and their applications: ICBMA 2019. Springer, 2021, pp. 39–50.

[18] F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. M. Chaabani, and A. Taleb-Ahmed, “Network intrusion detection system using neural network and condensed nearest neighbors with selection of nsl-kdd influencing features,” in 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS). IEEE, 2021, pp. 23–29.

[19] M HASSAN ZAIB, “NSL-KDD — Kaggle.” [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd

[20] E. Bisong and E. Bisong, “Introduction to scikit-learn,” Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, pp. 215–229, 2019.

[21] A. Pashamokhtari, G. Batista, and H. H. Gharakheili, “Adiotack: Quantifying and refining resilience of decision tree ensemble inference models against adversarial volumetric attacks on iot networks,” Computers & Security, vol. 120, p. 102801, 2022.

[22] S. Tufail, S. Batool, and A. I. Sarwat, “A comparative study of binary class logistic regression and shallow neural network for ddos attack prediction,” in SoutheastCon 2022. IEEE, 2022, pp. 310–315.

[23] A. Raza, H. U. R. Siddiqui, K. Munir, M. Almutairi, F. Rustam, and I. Ashraf, “Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction,” Plos one, vol. 17, no. 11, p. e0276525, 2022.

[24] S. Ismail and H. Reza, “Evaluation of na¨ıve bayesian algorithms for cyber-attacks detection in wireless sensor networks,” in 2022 IEEE World AI IoT Congress (AIIoT). IEEE, 2022, pp. 283–289.

[25] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, “Intrusion detection system combined enhanced random forest with smote algorithm,” EURASIP Journal on Advances in Signal Processing, vol. 2022, no. 1, pp. 1–20, 2022.

[26] F. Rustam, M. F. Mushtaq, A. Hamza, M. S. Farooq, A. D. Jurcut, and I. Ashraf, “Denial of service attack classification using machine learning with multi-features,” Electronics, vol. 11, no. 22, p. 3817, 2022.

[27] S. Kaur and M. Singh, “Hybrid intrusion detection and signature generation using deep recurrent neural networks,” Neural Computing and Applications, vol. 32, pp. 7859–7877, 2020.

[28] S. M. Kasongo and Y. Sun, “A deep gated recurrent unit based model for wireless intrusion detection system,” ICT Express, vol. 7, no. 1, pp. 81–87, 2021.

[29] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, “Intrusion detection systems using long short-term

memory (lstm),” Journal of Big Data, vol. 8, no. 1, p. 65, 2021.

[30] Y. Lin, H. Zhao, X. Ma, Y. Tu, and M. Wang, “Adversarial attacks in modulation recognition with convolutional neural networks,” IEEE Transactions on Reliability, vol. 70, no. 1, pp. 389–401, 2020.

[31] S. Zargar, “Introduction to sequence learning models: Rnn, lstm, gru,” no. April, 2021.

[32] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” Journal of Machine Learning Research, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[33] G.Viswanath, “Hybrid encryption framework for securing big data storage in multi-cloud environment”, Evolutionary intelligence, vol.14, 2021, pp.691-698.

[34] Viswanath Gudditi, “Adaptive Light Weight Encryption Algorithm for Securing Multi-Cloud Storage”, Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol.12, 2021, pp.545-552.

[35] Viswanath Gudditi, “A Smart Recommendation System for Medicine using Intelligent NLP Techniques”, 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), 2022, pp.1081-1084.

[36] G.Viswanath, “Enhancing power unbiased cooperative media access control protocol in manets”, International Journal of Engineering Inventions, 2014, vol.4, pp.8-12.

[37] Viswanath G, “A Hybrid Particle Swarm Optimization and C4.5 for Network Intrusion Detection and Prevention System”, 2024, International Journal of Computing, DOI: https://doi.org/10.47839/ijc.23.1.3442, vol.23, 2024, pp.109-115.

[38] G.Viswanath, “A Real Time online Food Ording application based DJANGO Restfull Framework”, Juni Khyat, vol.13, 2023, pp.154-162.

[39] Gudditi Viswanath, “Distributed Utility-Based Energy Efficient Cooperative Medium Access Control in MANETS”, 2014, International Journal of Engineering Inventions, vol.4, pp.08-12.

[40]G.Viswanath,“ A Real-Time Video Based Vehicle Classification, Detection And Counting System”, 2023, Industrial Engineering Journal, vol.52, pp.474-480.

[41] G.Viswanath, “A Real- Time Case Scenario Based On Url Phishing Detection Through Login Urls ”, 2023, Material Science Technology, vol.22, pp.103-108.

[42] Manmohan Singh,Susheel Kumar Tiwari, G. Swapna, Kirti Verma, Vikas Prasad, Vinod Patidar, Dharmendra Sharma and Hemant Mewada, “A Drug-Target Interaction Prediction Based on Supervised Probabilistic Classification” published in Journal of Computer Science, Available at: https://pdfs.semanticscholar.org/69ac/f07f2e756b79181e4f1e75f9e0f275a56b8e.pdf