Utilitas Mathematica
ISSN: 0315-3681 Volume 122, 2025

DEEP ENSEMBLE-BASED EFFICIENT FRAMEWORK FOR NETWORK ATTACK DETECTION

Devi Kumari Purre, Subramanyam Kodukula,

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur 522302, AP, India.

Abstract: Networks are essential for numerous sports, including corporate operations, instructional hobbies, and each day long-distance conversation. Although networks provide several advantages, additionally they pose safety risks that can jeopardise statistics confidentiality, integrity, and privacy. Network threats, including virus, hacking, and phishing, are increasing, leading to enormous economic and reputational harm. The assignment proposes the introduction of an automatic system utilizing artificial intelligence (AI) to mitigate those protection vulnerabilities. This generation is designed to effectively become aware of and protect in opposition to community threats, for this reason improving the security of statistics and networked structures. The mission implements an ensemble model that integrates 3 deep learning architectures: LSTM, RNN, and GRU. These models collaborate employing majority balloting standards to get extended accuracy inside the identification of network attacks, therefore making sure strong safety for networked environments. The venture augments its ability through the incorporation of a "Voting Classifier (Random Forest AdaBoost) and a Stacking Classifier", the latter achieving an amazing a 100% accuracy, which demonstrates improved effectiveness in detecting community attacks.

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

1. INTRODUCTION

The network suggests the interconnection of more than one computing gadgets and lets in for data sharing. information can be shared the use of distinctive technologies and conversation protocols, consisting of Ethernet, or easy cable connectivity [1]. The main cause of the community is to enable the coordination of gadgets and sharing of resources, which include printers, wireless servers and net connectivity. In modern-day interdependent global, networks are required for trade, mastering and day by day existence, making communication and records sharing a simple procedure over considerable distances. The community can provide numerous wireless like sharing sources, higher verbal exchange, pooled facts, cooperation, greater productivity, scalability, and faraway get right of entry to [2].

Many community programs can pose wi-fi threats and security safety and accordingly compromise the con wireless dentiality, integrity and availability of network structures and facts [3]. normal community threats are malware, hacking, phishing, DOS, MITM and spoofing. particular community assault threats are

facts theft, system damage, harm to reputation, wi-fi loss, espionage and infrastructure damage [4]. The increase in network attacks heightened the call for an automated device of detecting attacks. AI -powered solution has the functionality to hit upon such assaults, which makes it simpler to limit the threat of records robbery. Such techniques are utilized to test massive community records and become aware of ability threats in actual time, permitting companies to respond hastily and wi-fi spaciously wireless.

gadget studying techniques pick out styles from facts and are applied to become aware of future attacks. Incorporating these measures into network security can go a protracted manner in enhancing the capability of the business enterprise to identify and react to threats, making a hit violation less probably and making sure crucial data and assets. community safety is needed for organizations to safeguard the con wireless dentiality and privacy of their facts. because of this, a large quantity of research papers exist on community security. therefore, some other related works are discussed. In [7] a gadget of community intrusion makes use of based totally on conventional device learning strategies is designed.

system learning algorithms are carried out to the NSL-KDD information wireless. The end result shows that tree-based totally procedures provide wireless performance in community detection. The XGBOOST suggested attains 97% accuracy in wireless the assault. Likewise, the wireless of community disruptions with neural networks is wi-fi in [8] making use of the NSL-KDD facts wi-file wireless. Experimental results exhibit that the 2 -way LSTM method, popularized by the attention mechanism, plays very well.

2. LITERATURE SURVEY

Metavers is a hypothetical internet innovation that permits human beings to work, play and socially have interaction into a everlasting 3-D digital global and gives a fascinating enjoy via the development of a simulated universe that replicates reality, added by using real sounds, visible elements and other sensory additives. Metavers deposits stringent necessities for the completely immersive experience, adjusts more than one simultaneous customers and offers seamless connectivity, which is exceptional demanding situations for the 6th era wi-fi system (6G), such as ubiquitous connectivity, extremely-absorbent latency and reliability and reliability and strict security measures. furthermore, to assist a big population of users to have an engaging and trouble-loose experience, full-sensing, uninterrupted processing, strong cache storage, and enduring consensus and protection have to be very well included into the following 6G device. This paper objectives to describe the Roadmap to the Metavers on topics of conversation and Networking in 6G, including the Metavers Framework, Clarifying the Rigorous requirements and problems for 6G to Augalise The Metavers and Investigating the key technology to Be implemented in 6g to Facilitary The Implementation of the Metavers, which include smart Sensing, digital dual (DT), space-Air-floor-Sea included community (Sagsin), Multi-access side Computing (Mec), Blockchain and applicable protection problems.

After implementing 5G scientists and experts now expect the arrival of 6g. It is assumed that 6G will serve as the main catalyst for information exchange and social interaction after 2030. Using artificial intelligence (AI) will provide a 6G high -closed loop that will solve 5G deficiencies in communication,

calculation and worldwide coverage. In 6g, vehicles may appear as necessary gadgets for individuals along with smartphones to develop an illegal, highly safe and fully autonomous vehicles. In order to ensure the safe operation of future vehicles and dealing with the requirements for passenger entertainment, it is necessary to explore the future intelligence of 6G vehicles. This article will examine network, communication, computers and intelligence, explore future technological advances and applications and define imminent problems and research trajectories.

Detection of harmful operations on the Internet is an important challenge for academic workers aimed at protecting network infrastructure from harmful activities. The attacker can use several vulnerability in the network system to obtain unauthorized access through harmful communication. safety in opposition to such attacks needs to be an powerful automated device that could hit upon risky operation right now and forestall system harm. maximum computerized systems are capable of understand hazardous activities these days; however their effectiveness and precision need enhancement to apprehend dangerous communications from one of a kind area systems. This paintings focuses on precise identification modern day risky communications the use of machine learning state technique. [3] The advised methodology hired records sets: united states of america-NB15, which includes operating records primarily based on IoT and Iotid20, which has neighborhood community site visitors statistics. two data sets had been mixed to decorate the performance ultra-modern the cautioned approach for particular detection modern-day malicious operations from neighborhood and IoT networks. The horizontal merger trendy both datasets demands the equal wide variety modern-day functions that has been received by decreasing the range trendy features to 30 for each dataset document by means of thinking about the PCA. The cautioned model combines the stacked model called more Boosting woodland (EBF), which mixes tree-based totally models like greater bushes classifier, gradient random classifier and random forest based totally on a stacked record method. Empirical results imply that the EBF finished outstandingly and had a quality accuracy modern-day 0.985 and 0.984 on a multi-area information set for two and 4 classes.

At gift, the safety contemporary SDN is just like tips in traditional networks. but the nature present day such threats modifications through using a SDN [4]. Rejection modern-day service assault on the centralized controller manages the large network modern-day more than one devices (routers, switches, and so forth.) is greater detrimental in comparison to a specialized assault on an person router. The vulnerable SDN controller might permit the attacker to manipulate the entire community, whereas the compromised router could most effective intrude with the suitable functioning trendy the operation delegated to it. SDN will face new protection problems, especially inside the safeguarding present day SDN structure itself. security in a SDN is ensured at all tiers via three-layer layout and programming interfaces and constitutes multiple issues. The SDN protection problems will upward push while deployed. [4] This observe objectives to offer a complete description contemporary the equal time, classify research literature into taxonomy, highlighting the primary features and contributions modern-day concept for various layers present day SDN. Our literature evaluation highlights key studies gaps which could useful resource future research in this subject. The 6G community will likely supply real-time global connectivity and guide the transformation from

"related matters" to "connected intelligence," and community slicing is essential in network protection and delivering offerings for various and demanding scenarios contemporary vertical applications. Keeping with massive and sundry records, the solution based on artificial intelligence (AI) is greater apt than conventional models and algorithms for trouble-fixing that painting complicated and dynamic slicing in 6g. maintaining this in attitude, [5] here is a tutorial for an AI-aided community 6g [1, 2, 5] for slicing for securing the network and offering the offerings that is aimed to reveal the promise latest 6G reducing and advantage modern-day integrating the AI era. We discover six diverse attributes ultra-modern 6g modern day community cutting, verify the feasibility modern day AI in diverse network domains and technical elements, offer a case look at the bandwidth scaling with more AI and subsequently decide the key challenges and open problems for its future development.

3. METHODOLOGY

i) Proposed Work:

The method proposed employs a deep voting classifier (EDVC) to identify network attacks. The approach combines the predictions of three individual models of deep learning - LSTM, RNN and GRU - utilized the majority voting criterion to enhance the accuracy and reliability of network attack detection by consensus [8, 29, 31]. The astounding system's accuracy eliminates both missed attacks and false alarms and hence augments its credibility in identifying actual threats. The technology outnumbers current measures and demonstrates better capabilities for recognizing network threats as well as generalizing overall network protection.

The system has the ability to evaluate vast amounts of network information on the spot with efficiency in real time and promote swift counter responses to hazards as well as minimizing potential damages or threats. The project enhances its capability by combining the "voting classifier (Random Forest + Adaboost) and stacking classifier", the latter being one of the outstanding 100% accuracy and exhibits improved efficiency in network attack detection. An easy-to-use SQLite-integration flask is designed to enhance usability in real-world applications in cyber security, providing efficient registration procedures for user testing. This integration ensures a seamless and secure experience that offers the structure accessible and meaningful in real-life situations to enhance network security.

ii) System Architecture:

Figure 1 shows the proposed methodology architecture. An experimental data file containing network attack functions is utilized [9, 11, 12, 16]. Preliminary processing of the data file involves target attack mapping and categorical encoding of functions. Analysis of patterns of network attack functions is applied using reconnaissance data. The data are split into a training kit and a test kit in the proportion 0.8 to 0.2 for experiments. The attack detection method as proposed is applied to detect network attacks such as "normal, DOS, remote-to-local (R2L), probes and user-to-root (U2R) attacks".

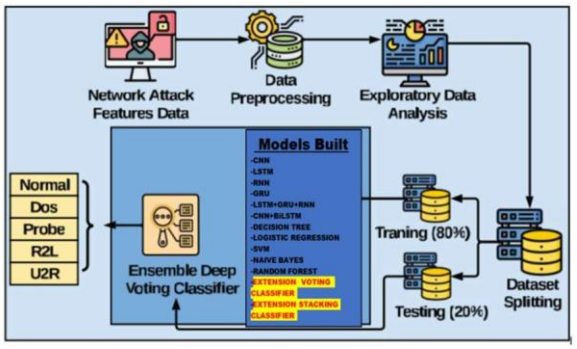


Fig 1: Proposed architecture

This structure guarantees that the project computes data analysis, trains advanced models, and employs file methods in accurately identifying network attacks by categories.

iii) Dataset collection:

A NSLKDD Benchmark data file is used, which contains publicly available network attack data [19]. Data set characteristics are derived from network attacks DOS, R2L, probes and U2R. The collection includes 148,517 items and 43 attributes on network attacks. Machine learning models work on numerical data; Our data set containing categorical functions requires pre -processing for disinfecting and transforming data into numerical format before the model input. We overtake the data file by deleting duplicate items and coding categorical variables. The "protocol type", "Services" and "Symptoms" feature were encoded with the Label encoder module within Scikit-Learn [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 is the conversion of raw facts to beneficial facts for organizations. data scientists generally undergo records processing, encompassing series, enterprise, cleansing, validation, evaluation and records transformation into understandable forms like graphs or reviews. statistics processing can be performed via three approaches: guide, mechanical, and digital. The purpose is to decorate the value of data and make selections extra efficient. This helps organizations to strengthen their operations and take fast strategic choices. automatic data processing equipment, which include pc programming for software, are important in such instances. It has the capacity to transform crucial amounts of statistics, mainly heavy records, to vital understanding for high-quality and decision -making.

v) Feature selection:

The feature selection is the assignment of selecting the maximum conteable, non -re -dundant and maximum beautiful traits for building the model. Systematic discount of the statistics set's dimensions is needed, despite the fact that data set length and diversity keep growing. the main goal in selecting factors is to maximize the predictive version's efficiency and, at the identical time, lessen the modeling computing costs.

The deciding on of capabilities, the essential belongings of useful engineering accommodates identification of the most important capabilities for inputs of the machine learning algorithms. The method for choosing is utilized to lower the quantity of enter variables with the aid of the removal of useless or redundant functions, and as such complements the set to the maximum suitable to the device learning model. the primary blessings of choosing functions in advance, as opposed to letting the device mastering model decide on the most important capabilities robotically.

vi) Algorithms:

"CNN (Convolutional Neural Network)"- CNN is a DL to know architecture that is normally used for photo analysis. This undertaking may be changed to extraction of community information factors, detects formulas or abnormalities within the spatial facts shape [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)" - LSTM is a variation of "recurrent neural networks (RNN)" identified as its potential to technique sequential records. It is effective in detecting formulas in network facts of the time series, that's beneficial for looking at the behavior of the assault that manifests through the years [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)- RNN is every other deep architecture of mastering for the look at of collection records. It can become aware of formulation and dependencies in community records sequences, which enables community assaults with time attributes [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)" - GRU is a variant of RNN, which streamlines structure and on the same time keeps the capability to capture sequential patterns. It efficaciously models dependence in

community information, which makes it less complicated to hit upon quickly orientated attacks [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" - This file integrates the benefits of "LSTM, GRU and RNN" and gives a more long lasting approach for detecting network assaults. The use of diverse sorts of recurring neural networks lets in effective seize of various formulas in community facts.

```
# 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(model.summary())
```

Fig 7: LSTM+GRU+RNN

"CNN+BiLSTM (Bidirectional LSTM)" - This integration of CNN and Bilstm makes it less complicated to research extraction and sequence. It can apprehend spatial and time styles in community information, which will increase the potential to locate complex assaults.

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

model2 = tf.keras.models.Sequential([tf.keras.layers.Conv1D(filters=128,kernel_size=5,strides=1,padding="causal",activation="reli
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, padding="valid"),
    tf.keras.layers.Conv1D(filters=64, kernel_size=3, strides=1, padding="causal", activation="relu"),
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, padding="valid"),
    tf.keras.layers.Conv1D(filters=32, kernel_size=3, strides=1, padding="causal", activation="relu"),
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, 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(0.2),
    tf.keras.layers.Dense(32, activation="relu"),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(5)
])

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(5e-4,
                                                             decay_steps=1000000,
                                                             decay_rate=0.98,
                                                             staircase=False)

model2.compile(loss=tf.keras.losses.MeanSquaredError(),
               optimizer=tf.keras.optimizers.SGD(learning_rate=lr_schedule, momentum=0.8),
               metrics=['acc'])
```

Fig 8: CNN + BiLSTM

Decision Tree - DT represent the method beneath supervision. The undertaking can use them to expand a selection -making version to discover network attacks based totally on attributes and their relationships.

```
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" – LR is a statistical model often used for binary classification. In this case, it can help in categorizing network data as normal or instance of attack [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)" - SVM guide is a device ML of method that categorizes records into discrete training. It is critical to differentiate network attacks from popular network visitors [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 - Naive Bayes is a probability technique used for classification purposes. It can calculate the likelihood of network data belonging to several categories (normal or attack), which makes it easier to classify.

```
# 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" – "Random Forest" is a technique of mastering a document that integrates a number of DT to enhance the accuracy of type. It may be used to broaden a extra resistant version to discover community threats [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 - The voting classifier merges predictions from various machine learning models. This is used to consolidate knowledge from several models, which increases the overall accuracy of the attack detection.

```
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 integrates predictions of different fashions using a meta-classifier. It improves a community attack detection system by way of thinking about numerous version outputs and imparting a more correct very last end.

```
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: The precision evaluates the share of exactly categorized instances amongst cases recognized as wonderful. As a end result, the formulation for calculating precision is expressed:

"Precision = True positives/ (True positives + False positives) = TP/(TP + FP)"

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

Fig 16: Precision comparison graph

Recall: The recall is a meter in ML that evaluates the potential of the model to understand all applicable instances of a particular class. It is the ratio of precisely anticipated effective observations to universal real positives and gives perception into the version performance in figuring out the incidence of a selected magnificence.

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

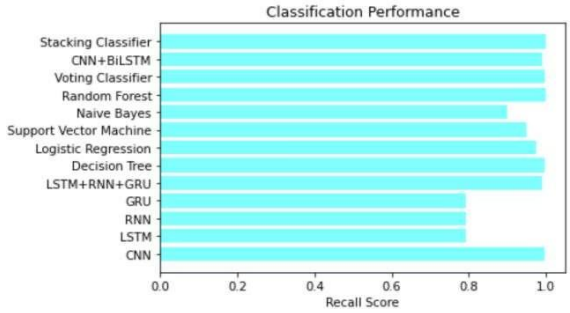


Fig 17: Recall comparison graph

Accuracy: The accuracy suggests the ratio of the ideal predictions inside the class undertaking and evaluates the general accuracy of the model predictions.

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

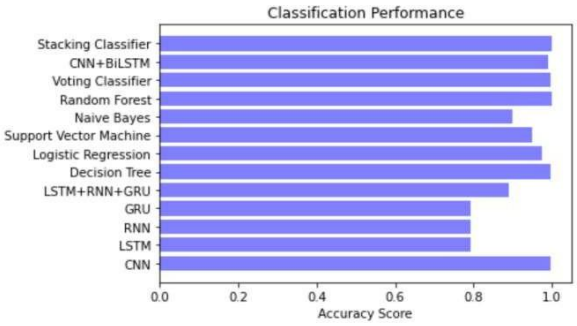


Fig 18: Accuracy graph

F1 Score: The F1 Score is a harmonious diameter of precision and recall that offers a balanced metric this is chargeable for false positives and false negatives, that's appropriate for unbalanced data sets.

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

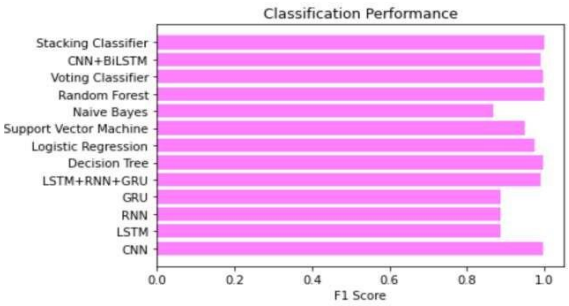


Fig 19: F1Score

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	0.990	0.991
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

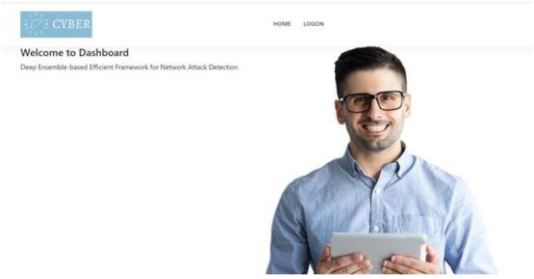


Fig 21: Home page

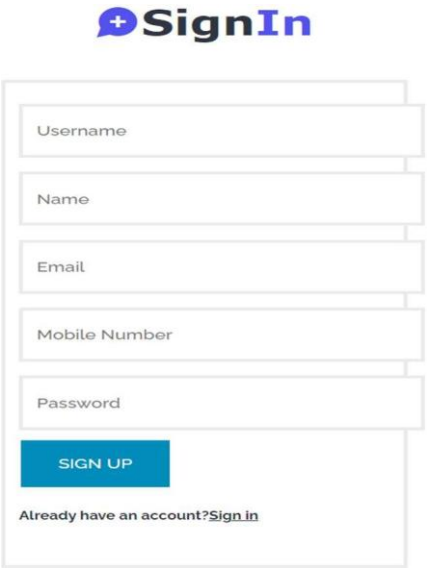
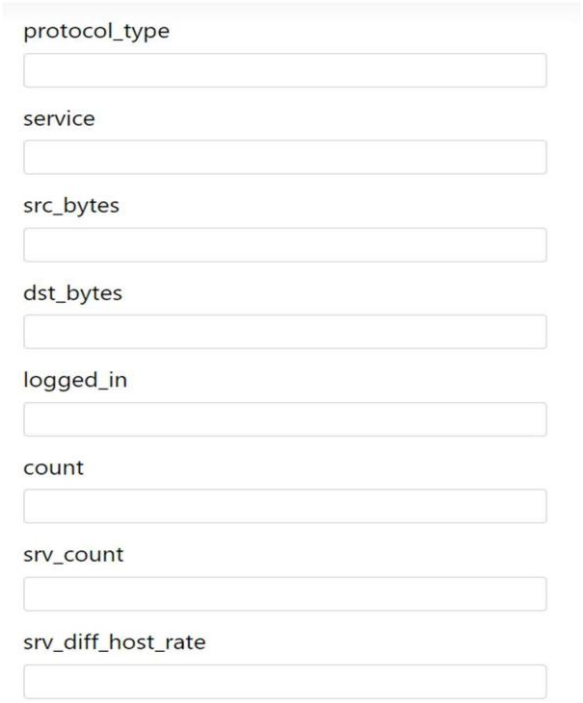


Fig 22: Signin page

Fig 23: Login page



"Fig 24 User input"

Fig 25: Predict result for given input

5. CONCLUSION

Research strategically integrates state-of-the-art deep learning to know fashions, along with LSTM, RNN, and GRU to network attack detection [27, 29]. The challenge seeks to enhance the safety of the computer network by using the characteristics of those models and gives robust defense against diverse network threats. The project body is challenge to an intensive evaluation the use of the NSL-KDD information set, which demonstrates its efficiency in correctly figuring out the network hazard. This empirical validation illustrates the sensible usefulness of the framework and its capacity to efficaciously detect and reply to harmful events in pc networks. The integration of deep gaining knowledge of fashions and report techniques endorses the framework with flexibility, permitting it to analyze and adapt to changing assault patterns. This adaptability is essential to enhance the resistance of the network to emerging attacks, which guarantees that the framework remains effective in fixing the evolving nature of concerns approximately cyber safety. The assignment increases its competencies with the aid of the use of file procedures, which include the voting classifier and stacking classifier, main to an development in the accuracy of community assault detection. Incorporating an intuitive flask interface with robust authentication improves the consumer enjoy at some stage in the gadget trying out. This interface permits you to enter information for overall performance assessment and draws the frame accessible and practical for cyber protection specialists. The integration of report and intuitive features improves the efficiency and practicality of the frame inside the actual global programs.

6. FUTURE SCOPE

Future effort will focus on strengthening the computing efficiency of the system by way of exploring the structure of particular deep studying fashions. This optimization will improve network assault detection performance, as a way to be strong and economic. Emphasizing the scalability of the machine will increase its practicality for implementation in real community protection solutions. This improve ensures that the machine can control large and greater complicated networks whilst lowering operating prices. The upcoming mission tasks include the creation of equipment to come across network attacks in real time [19]. This functionality in real time allows the machine to speedy remedy and

alleviate risks when they rise up, minimizing viable harm and susceptibility. As the volumes of network information are increasing, the assignment will modify the framework to healthy sizeable network facts. This adaptation is crucial for controlling the growing complexity and quantity of community site visitors, which ensures the efficiency of the machine. Incorporating the capability to detect anomaly into the framework is an crucial a part of future efforts. This enhancement will permit the gadget to understand new and developing assault styles and offer a proactive network safety method and reply to threats.

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