International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

AN INTRUSION DETECTION METHOD BASED ON TRANSFORMER-LSTM MODEL

N. MOULALI, Assistant Professor, Dept of MCA, Chirala Engineering College, Chirala, noorbashamoulali@gmail.com

POGADADANDA NAGA SIVAKANTH, PG Student - MCA, Dept of MCA, Chirala Engineering College, Chirala, sivakanth0707@gmail.com

Abstract: The project focuses on addressing the escalating issue of web attacks and ensuring cyberspace security in the era of advancing internet technology. The primary goal is to enhance intrusion detection systems (IDS) by moving beyond traditional keyword and rule-based methods. The project explores the integration of artificial intelligence, specifically deep learning techniques, to develop a more robust and adaptive approach to identifying network intrusion events. With the limitations of conventional detection methods in mind, there is a critical need for a more effective and versatile intrusion detection system. The rise in diverse attack methods necessitates an intelligent system capable of autonomously adapting to new attack patterns without relying on manual rule creation. This project is designed to benefit both individuals and organizations reliant on network services. Improved intrusion detection ensures a higher level of cybersecurity, safeguarding sensitive data and systems. The project's outcomes have the potential to contribute significantly to the broader field of network security. Leveraging the power of deep learning, the proposed model- Transformer (autoencoder) LSTM combines Transformer and LSTM architectures. Multiple attention mechanisms are employed to select and extract features, while the LSTM model facilitates understanding the sequential relationships within network traffic, enabling the accurate identification of intrusion events. And we are comparing the model with DNN, LSTM and Transformer(autoencoder). To enhance performance, CNN and CNN+LSTM extensions were incorporated into the project, complementing the Transformer (autoencoder) LSTM model. These models aim to further refine feature extraction and exploit spatial relationships within network traffic.

Index Terms *- intrusion detection, transformer, LSTM, deep learning.*

1. INTRODUCTION

With the development of Internet technology, the Internet has more and more influence on people's work and life. Accordingly, the problem of cyberspace security has also become more important. Recently, network security issues, for example, web attacks, have become one of the most important issues in network security [1]. To ensure the security, Intrusion detection system (IDS) is widely studied.

Traditional network attack detection methods usually detect network attacks by matching keywords or rule bases. The rule bases contain the characteristics and description rules of known attack patterns. These detection methods detect attacks through regular matching. Although they can effectively detect existing network attack modes, they have great limitations. On the one hand, the intrusion detection method based on matching keywords or rule base can only identify the existing attack patterns in the rule base, and cannot respond to the new attack patterns [2,3]. Hackers can still easily bypass these rules by specific means. On the other hand, the construction and maintenance of rule base depends on security experts, which leads to its high maintenance cost and cannot be widely used. In addition, with the increasing diversity of attack methods, traditional intrusion detection methods are no longer sufficient to ensure the security of network services. It is necessary to propose more effective intrusion detection methods. In recent years, with the development of artificial intelligence technology, intrusion detection systems based on machine learning and deep learning methods have become thefocus of research.

[2] The intrusion detection method based on machine learning is more robust than the traditional intrusion detection method based on rule base or matching keywords, and also has good results in practical use, but this kind of method has quite high requirements for feature selection. Once the feature selection is inappropriate, the effect of intrusion detection system will be greatly reduced. The rules of intrusion detection based on deep learning are more sophisticated. By using the endto-end training process from the original traffic to the classification results, this kind of method can achieve the independent selection of appropriate traffic features in the training process, and can also achieve the extraction of complex features based on existing simple features. Therefore, the intrusion detection method based on deep learning is more universal than the traditional intrusion detection method based on machine learning [2].

There are many kinds of neural network models used in the existing deep learning, including some classical deep learning models, such as convolutional neural network (CNN) [5,9], recurrent neural network (RNN), and Transformer. These basic models have been widely used in various fields. Therefore, this paper proposes an intrusion detection model based on Transformer & LSTM [8,9]. The model uses multiple attention mechanism to complete the selection and extraction of features, and realizes the sequence relationship of traffic context through LSTM model, so as to identify network intrusion events.

2. LITERATURE SURVEY

Every day, security engineers cope with a flow of cyber security incidents. While most incidents trigger routine reactions, others require orders of magnitude more effort to investigate and resolve. How security operation teams in organizations should tune their response to tame extreme events remains unclear. Analyzing the statistical properties of sixty thousand security events collected over six years at a large organization, we find that the distribution of costs induced by security incidents is in general highly skewed, following a power law tail distribution [1]. However, this distribution of incident severity becomes less skewed over time, suggesting that the organization under scrutiny has managed to reduce the impact of large events. We illustrate this result with a case study focused on the empirical effects of full disk encryption on the severity of incidents involving lost or stolen devices.

Anomaly detection as a kind of intrusion detection is good at detecting the unknown attacks or new attacks, and it has attracted much attention during recent years. In this paper, a new hierarchy anomaly intrusion detection model that combines the fuzzy c-means (FCM) based on genetic algorithm and SVM is proposed [2,3,4]. During the process of detecting intrusion, the membership function and the fuzzy interval are applied to it, and the process is extended to soft classification from the previous hard classification [2]. Then a fuzzy error correction sub interval is introduced, so when the detection result of a data instance belongs to this range, the data will be re-detected in order to improve the effectiveness of intrusion detection. Experimental results show that the proposed model can effectively detect the vast majority of network attack types, which provides a feasible solution for solving the problems of false alarm rate and detection rate in anomaly intrusion detection model.

Intrusion detection is one essential tool towards building secure and trustworthy Cloud computing environment, given the ubiquitous presence of cyber attacks that proliferate rapidly and morph dynamically. In our current working paradigm of resource, platform and service consolidations, Cloud Computing provides a significant improvement in the cost metrics via dynamic provisioning of IT services [3]. Since almost all cloud computing networks lean on providing their services through Internet, they are prone to experience variety of security issues. Therefore, in cloud environments, it is necessary to deploy an Intrusion Detection System (IDS) to detect new and unknown attacks in addition to signature based known attacks, with high accuracy [3,5,6]. In our deliberation we assume that a system or a network “anomalous” event is synonymous to an “intrusion” event when there is a significant departure in one or more underlying system or network activities. There are couple of recently proposed ideas that aim to develop a hybrid detection mechanism, combining advantages of signature-based detection schemes with the ability to detect unknown attacks based on anomalies. In this work, we propose a network based anomaly detection system at the Cloud Hypervisor level that utilizes a hybrid algorithm: a combination of K-means clustering algorithm and SVM classification algorithm, to improve the accuracy of the anomaly detection system. Dataset from UNSW-NB15 study is used to evaluate the proposed approach and results are compared with previous studies. The accuracy for our proposed K-means clustering model is slightly higher than others. However, the accuracy we obtained from the SVM model is still low for supervised techniques.

A robust network intrusion detection system (NIDS) plays an important role in cyberspace security for protecting confidential systems from potential threats. In real world network, there exists complex correlations among the various types of network traffic information, which may be respectively attributed to different abnormal behaviors and should be make full utilized in NIDS [4]. Regarding complex network traffic information, traditional learning based abnormal behavior detection methods can hardly meet the requirements of the real world network environment. Existing methods have not taken into account the impact of various modalities of data, and the mutual support among different data features. To address the concerns, this paper proposes a multi-dimensional feature fusion and stacking ensemble mechanism (MFFSEM), which can detect abnormal behaviors effectively. In order to accurately explore the connotation of traffic information, multiple basic feature datasets are established considering different aspects of traffic information such as time, space, and load. Then, considering the association and correlation among the basic feature datasets, multiple comprehensive feature datasets are set up to meet the requirements of real world abnormal behavior detection [4]. In specific, stacking ensemble learning is conducted on multiple comprehensive feature datasets, and thus an effective multi-dimensional global anomaly detection model is accomplished. The experimental results on the dataset KDD Cup 99, NSL-KDD, UNSW-NB15, and CIC-IDS2017 have shown that MFFSEM significantly outperforms the basic and meta classifiers adopted in our method. Furthermore, its detection performance is superior to other well-known ensemble approaches.

Network intrusion detection system (NIDS) plays an important role in network security. It can detect the malicious traffic and prevent the network intrusion. Traditional methods used machine learning techniques such as support vector machine, Bayesian classification, decision tree and k-means. The traditional machine learning methods first need to manually select features and has obvious limitations. In this paper, we propose a novel NIDS system based on convolutional neural network. [5] We train deep-learning based detection models using both extracted features and original network traffic. We conduct

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

comprehensive experiments using well-known benchmark datasets. The results verify the effectiveness of our system and also demonstrate the model trained through raw traffic has better accuracy than the model trained using extracted features [3,5,6].

3. METHODOLOGY

i) Proposed Work:

In this project, we propose an innovative intrusion detection system leveraging the Transformer (autoencoder) LSTM model. Combining Transformer and LSTM architectures, our approach utilizes multiple attention mechanisms for feature selection and extraction. [7,9] The LSTM model enhances the understanding of sequential relationships within network traffic, enabling precise identification of intrusion events. We employ label encoding and Min-Max scaling in preprocessing, and feature selection is performed using SelectPercentile using mutual_info_classif. The proposed system is benchmarked against DNN, LSTM, and Transformer (autoencoder) for comprehensive evaluation and comparison.To enhance performance, CNN and CNN+LSTM models were incorporated into the project, complementing the Transformer (autoencoder) LSTM model. These models aim to further refine feature extraction and exploit spatial relationships within network traffic. Additionally, a Flask framework with SQLite was implemented for user testing, facilitating seamless signup and signin functionalities. This framework enables users to input data, receive results, and ensures a user-friendly experience for evaluating the extended intrusion detection system.

ii) System Architecture:

The system architecture begins with the KDDCup dataset input, undergoing data preprocessing and feature selection. The dataset is then split into training and testing sets. Model building involves diverse architectures, including DNN, LSTM, Transformer (autoencoder), Transformer (autoencoder) LSTM, CNN, and CNN+LSTM [6]. Performance evaluation metrics such as accuracy, precision, recall, and F1 score are employed. The final stage encompasses attack classification and detection, forming a comprehensive and versatile intrusion detection system.

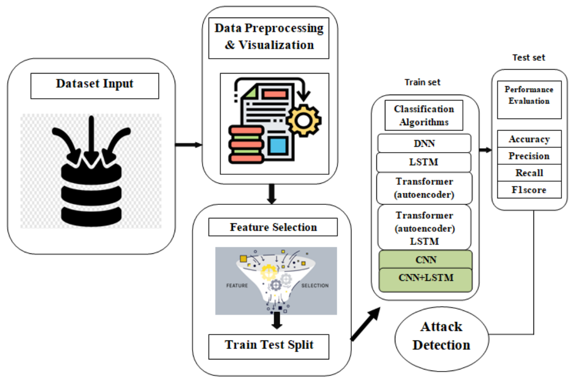


Fig 1 Proposed architecture

International journal of basic and applied research

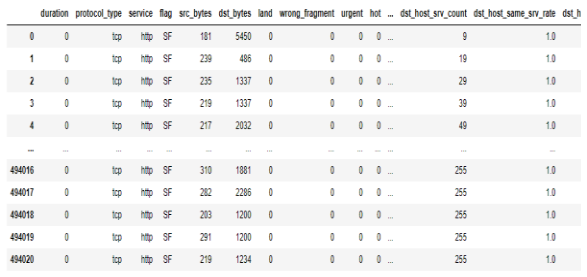
www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

iii) Dataset collection:

The KDD Cup dataset is a benchmark dataset widely used in intrusion detection research. It originated from the KDD Cup 1999 competition and contains network traffic data, providing a diverse and representative sample for evaluating intrusion detection system performance [13].



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_h
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0	
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0	
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0	
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0	
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
494016	0	tcp	http	SF	310	1881	0	0	0	0	...	255	1.0	
494017	0	tcp	http	SF	282	2286	0	0	0	0	...	255	1.0	
494018	0	tcp	http	SF	203	1200	0	0	0	0	...	255	1.0	
494019	0	tcp	http	SF	291	1200	0	0	0	0	...	255	1.0	
494020	0	tcp	http	SF	219	1234	0	0	0	0	...	255	1.0	

Fig 2 Dataset

iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

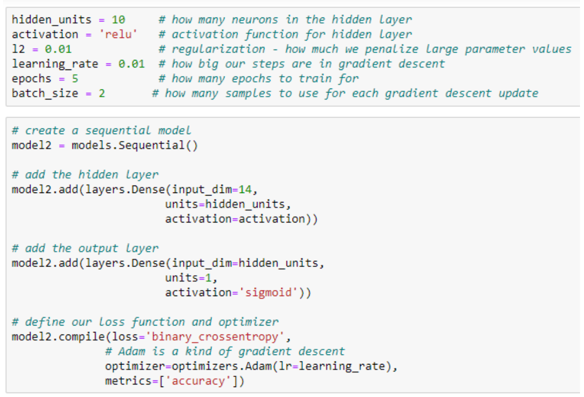
v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

Deep Neural Network (DNN) is a multi-layered neural network architecture designed for intricate pattern extraction. Comprising input, hidden, and output layers, DNN excels at learning complex relationships within data. In this project, DNN is selected to effectively capture nuanced patterns in network traffic. Its hierarchical structure facilitates automatic feature extraction, making it a powerful tool for robust intrusion detection in the dynamic and complex landscape of cybersecurity.



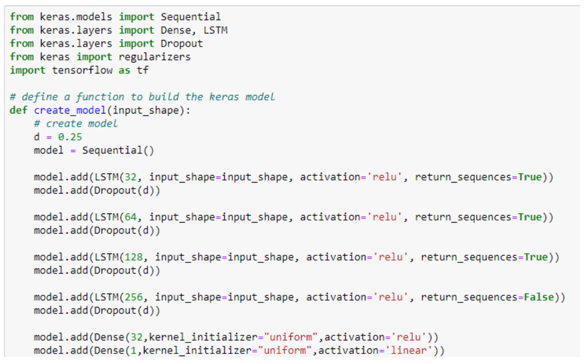
```
hidden_units = 10     # how many neurons in the hidden layer
activation = 'relu'   # activation function for hidden layer
l2 = 0.01             # regularization - how much we penalize large parameter values
learning_rate = 0.01  # how big our steps are in gradient descent
epochs = 5            # how many epochs to train for
batch_size = 2       # how many samples to use for each gradient descent update
```

```
# create a sequential model
model2 = models.Sequential()

# add the hidden layer
model2.add(layers.Dense(input_dim=14,
                        units=hidden_units,
                        activation=activation))

# add the output layer
model2.add(layers.Dense(input_dim=hidden_units,
                        units=1,
                        activation='sigmoid'))

# define our loss function and optimizer
model2.compile(loss='binary_crossentropy',
               # Adam is a kind of gradient descent
               optimizer=optimizers.Adam(lr=learning_rate),
               metrics=['accuracy'])
```

Fig 3 DNN

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to address the vanishing gradient problem in traditional RNNs. LSTMs have a unique architecture with memory cells and gates, allowing them to selectively remember and forget information over extended sequences. In this project, LSTM is utilized for its ability to capture sequential patterns in network traffic data, crucial for effective intrusion detection where the sequence relationships within traffic are essential for accurate classification [7,9].

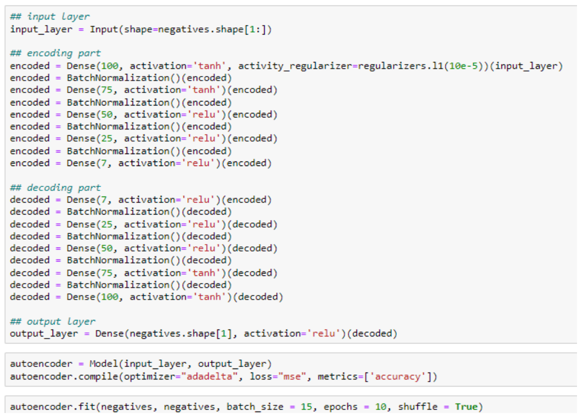
LSTM



```
from keras.models import Sequential
from keras.layers import Dense, LSTM
from keras.layers import Dropout
from keras import regularizers
import tensorflow as tf

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

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

    model.add(LSTM(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'))
```

Fig 4 LSTM

The Transformer (Autoencoder) is a neural network architecture that utilizes self-attention mechanisms for feature extraction. In the context of this project, it employs an autoencoder structure, reconstructing input data and learning robust representations. The self-attention mechanism allows the model to focus on relevant features, making it suitable for capturing intricate patterns in network traffic data, crucial for effective intrusion detection.



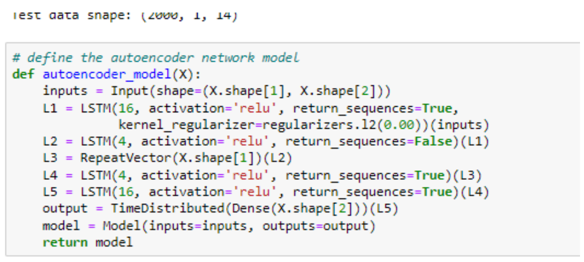
```
## input layer
input_layer = Input(shape=negatives.shape[1:])

## encoding part
encoded = Dense(100, activation='tanh', activity_regularizer=regularizers.l1(10e-5))(input_layer)
encoded = BatchNormalization()(encoded)
encoded = Dense(75, activation='tanh')(encoded)
encoded = BatchNormalization()(encoded)
encoded = Dense(50, activation='relu')(encoded)
encoded = BatchNormalization()(encoded)
encoded = Dense(25, activation='relu')(encoded)
encoded = BatchNormalization()(encoded)
encoded = Dense(7, activation='relu')(encoded)

## decoding part
decoded = Dense(7, activation='relu')(encoded)
decoded = BatchNormalization()(decoded)
decoded = Dense(25, activation='relu')(decoded)
decoded = BatchNormalization()(decoded)
decoded = Dense(50, activation='relu')(decoded)
decoded = BatchNormalization()(decoded)
decoded = Dense(75, activation='tanh')(decoded)
decoded = BatchNormalization()(decoded)
decoded = Dense(100, activation='tanh')(decoded)

## output layer
output_layer = Dense(negatives.shape[1], activation='relu')(decoded)
```

```
autoencoder = Model(input_layer, output_layer)
autoencoder.compile(optimizer="adadelta", loss="mse", metrics=['accuracy'])
```

```
autoencoder.fit(negatives, negatives, batch_size = 15, epochs = 10, shuffle = True)
```

Fig 5 Transformer (Autoencoder)

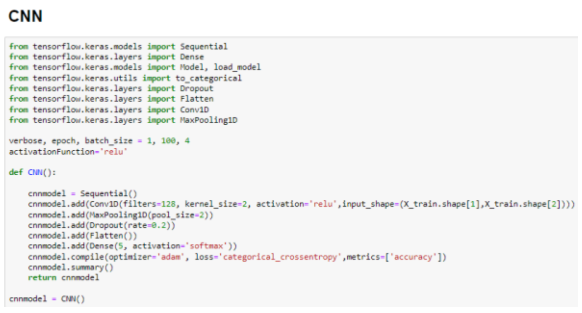
The Transformer (Autoencoder) LSTM is a hybrid model combining Transformer and LSTM architectures. The model employs multiple attention mechanisms for feature selection and extraction. The Transformer's ability to capture global relationships in data is complemented by LSTM's sequential understanding. This combination enhances the model's capacity to identify complex patterns in network traffic, making it suitable for the diverse and evolving nature of intrusion detection in this project.



```
Test data shape: (2000, 1, 14)

# define the autoencoder network model
def autoencoder_model(X):
    inputs = Input(shape=(X.shape[1], X.shape[2]))
    L1 = LSTM(16, activation='relu', return_sequences=True,
              kernel_regularizer=regularizers.l2(0.00))(inputs)
    L2 = LSTM(4, activation='relu', return_sequences=False)(L1)
    L3 = RepeatVector(X.shape[1])(L2)
    L4 = LSTM(4, activation='relu', return_sequences=True)(L3)
    L5 = LSTM(16, activation='relu', return_sequences=True)(L4)
    output = TimeDistributed(Dense(X.shape[2]))(L5)
    model = Model(inputs=inputs, outputs=output)
    return model
```

Fig 6 Transformer (Autoencoder) LSTM

Convolutional Neural Networks (CNNs) are deep learning models designed for processing structured grid data, such as images. They consist of convolutional layers that apply filters to input data, capturing spatial hierarchies. This project leverages CNNs for their ability to automatically extract relevant features from network traffic data, aiding in effective intrusion detection. The model's hierarchical structure and shared weight architecture make it well-suited for capturing patterns in sequential data, aligning with the diverse characteristics of network traffic for accurate intrusion detection.

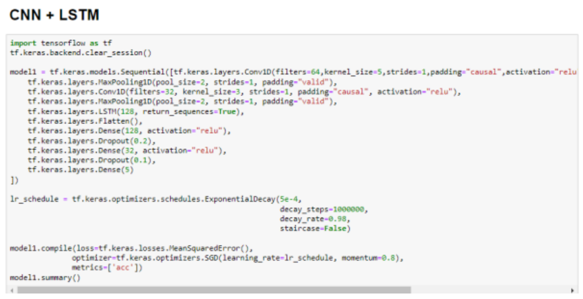


CNN

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import MaxPooling1D

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_train.shape[2])))
    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 7 CNN

CNN+LSTM, a hybrid model, combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. CNN excels in spatial feature extraction from data, while LSTM specializes in capturing temporal dependencies. In this project, CNN+LSTM is chosen for its ability to effectively analyze both spatial and sequential relationships within network traffic data, enhancing the intrusion detection system's overall performance. The CNN+LSTM structure allows for comprehensive feature extraction and sequence learning, making it well-suited for capturing intricate patterns indicative of network intrusions.



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

model1 = tf.keras.models.Sequential([tf.keras.layers.Conv1D(filters=64,kernel_size=5,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.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)

model1.compile(loss=tf.keras.losses.MeanSquaredError(),
              optimizer=tf.keras.optimizers.SGD(learning_rate=lr_schedule, momentum=0.8),
              metrics=['acc'])
model1.summary()
```

Fig 8 CNN + LSTM

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

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

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

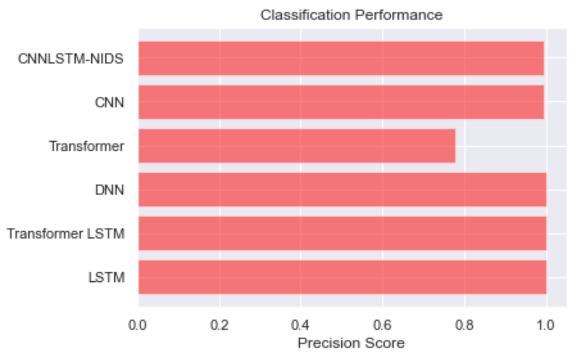


Fig 9 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

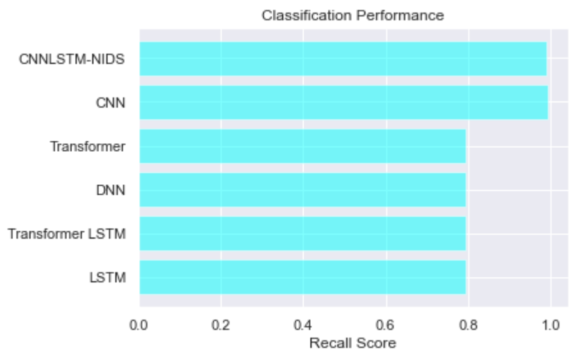


Fig 10 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

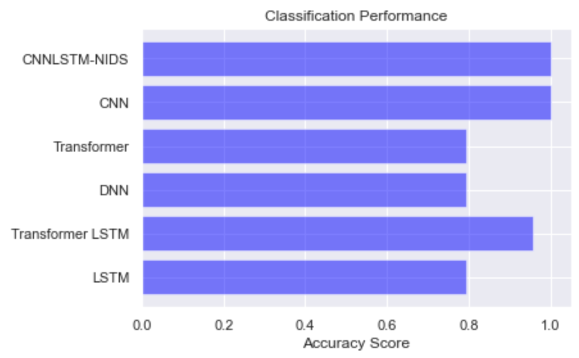


Fig 11 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

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

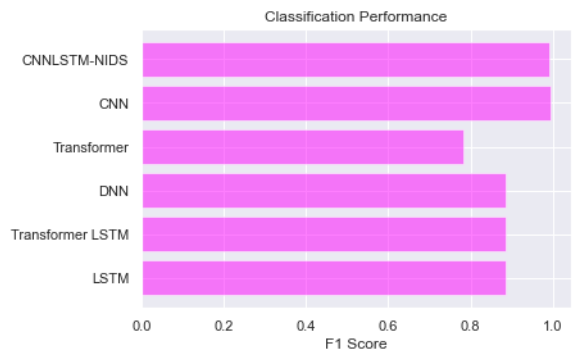
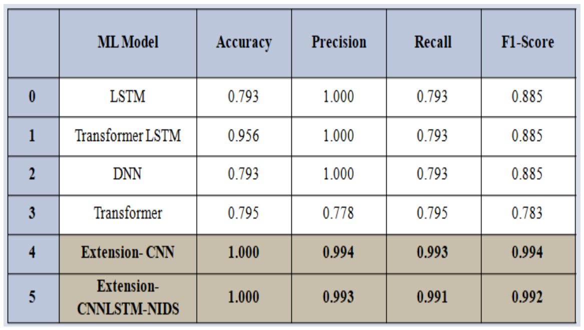
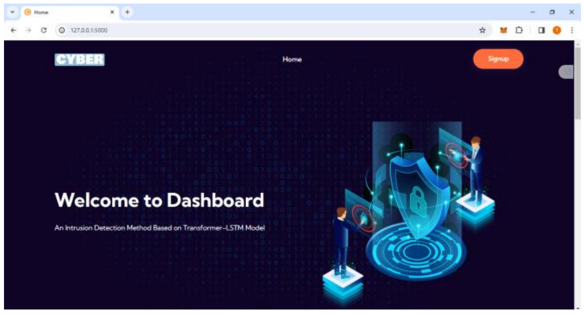


Fig 12 F1Score



	ML Model	Accuracy	Precision	Recall	F1-Score
0	LSTM	0.793	1.000	0.793	0.885
1	Transformer LSTM	0.956	1.000	0.793	0.885
2	DNN	0.793	1.000	0.793	0.885
3	Transformer	0.795	0.778	0.795	0.783
4	**Extension- CNN**	**1.000**	**0.994**	**0.993**	**0.994**
5	**Extension-CNNLSTM-NIDS**	**1.000**	**0.993**	**0.991**	**0.992**

Fig 13 Performance Evaluation



Welcome to Dashboard

An Intrusion Detection Method Based on Transformer-LSTM Model

Fig 14 Home page

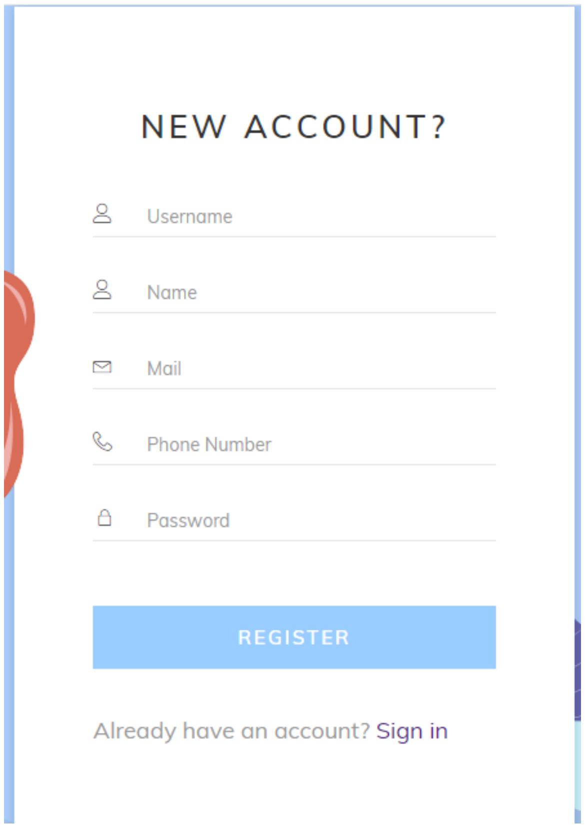


Fig 15 Signin page

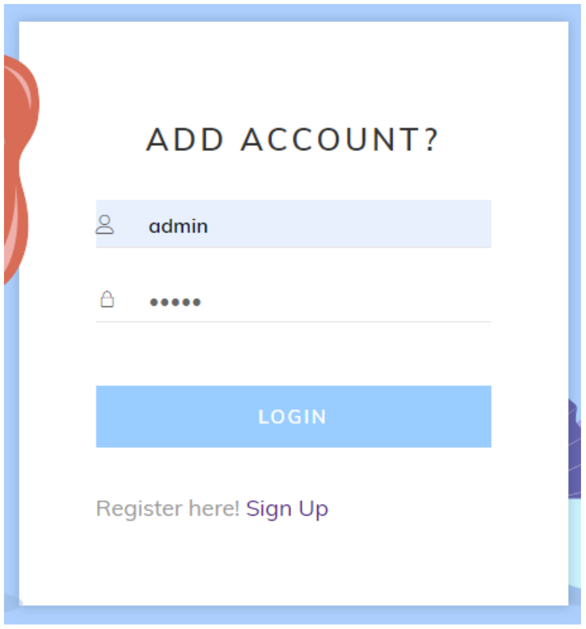
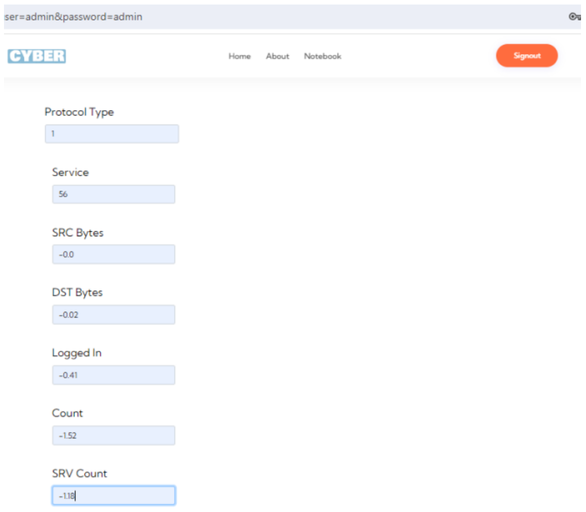


Fig 16 Login page



ser=admin&password=admin

Protocol Type: 1

Service: 56

SRC Bytes: -0.0

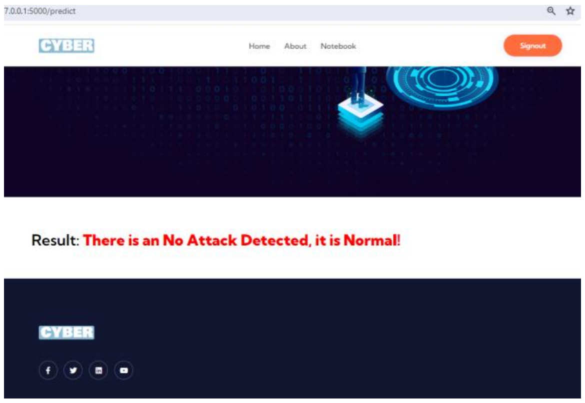
DST Bytes: -0.02

Logged In: -0.41

Count: -1.52

SRV Count: -1.18

Fig 17User input



Result: There is an No Attack Detected, it is Normal!

Fig 18 Predict result for given input

5. CONCLUSION

This project pioneers enhanced intrusion detection through deep learning, leveraging meticulous data processing with Pandas and Keras to optimize the KDD Cup dataset, elevating the overall security posture. Thoroughly exploring DNN, LSTM, Transformer (Autoencoder), and hybrid model (Transformer(Autoencoder) LSTM) aims to identify the most effective intrusion detection approach, ensuring a comprehensive and adaptive system. Implementing CNN and CNN+LSTM as extensions improves prediction accuracy, fortifying the intrusion detection system's reliability [7,9]. The added models enhance feature extraction and spatial relationship analysis, contributing to superior performance. Integrating Flask and SQLite creates a user-friendly interface, fostering broader accessibility. The design prioritizes user testing, input validation, and seamless model predictions, enhancing practical usability and encouraging wider adoption. Real-time features and adaptive systems showcase a proactive threat response for heightened cybersecurity. This project significantly benefits cybersecurity analysts, organizations, and individuals reliant on network services. The user-friendly interface and adaptive system design cater to diverse users, ensuring improved intrusion detection for heightened cybersecurity across various contexts.

6. FUTURE SCOPE

Future works could include, exploring the incorporation of adversarial training techniques to enhance the model's robustness against sophisticated attacks, ensuring a more resilient intrusion detection system. Investigating real-time deployment capabilities, optimizing the model for on-the-fly analysis of network traffic, allowing for immediate detection and response to emerging threats. Developing dynamic feature selection mechanisms to adaptively choose relevant features based on evolving attack patterns, ensuring the model's adaptability to emerging cyber threats without manual intervention. Focusing on improving the interpretability of the model's decisions, employing techniques such as attention mechanisms and explainable AI, making the intrusion detection system more transparent and understandable for cybersecurity analysts [3,5].

REFERENCES

[1] Kuypers, M. A., Maillart, T., & Paté-Cornell, E. (2016). An empirical analysis of cyber security incidents at a large organization. Department of Management Science and Engineering, Stanford University, School of Information, 30. https://fsi-live.s3.us-west-1.amazonaws.com/s3fspublic/kuypersweis_v7.pdf

[2] Tang, C., Xiang, Y., Wang, Y., Qian, J., & Qiang, B. (2016). Detection and classification of anomaly intrusion using hierarchy clustering and SVM. Security and Communication Networks, 9(16), 3401-3411. https://doi.org/10.1002/sec.1547

[3] Mohamad Tahir, H., Hasan, W., Md Said, A., Zakaria, N. H., Katuk, N., Kabir, N. F., ... & Yahya, N. I. (2015). Hybrid machine learning technique for intrusion detection system. https://repo.uum.edu.my/id/eprint/15600

[4] Zhang, H., Li, J. L., Liu, X. M., & Dong, C. (2021). Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. Future Generation Computer Systems, 122, 130-143. https://doi.org/10.1016/j.future.2021.03.024

[5] Chen, L., Kuang, X., Xu, A., Suo, S., & Yang, Y. (2020, December). A novel network intrusion detection system based on CNN. In 2020 eighth international conference on advanced cloud and big data (CBD) (pp. 243- 247). IEEE. https://doi.org/10.1109/CBD51900.2020.00051

[6] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089. https://doi.org/10.48550/arXiv.1802.09089

[7] Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. Expert Systems with Applications, 185, 115524. https://doi.org/10.1016/j.eswa.2021.115524

[8] Yang, Y. G., Fu, H. M., Gao, S., Zhou, Y. H., & Shi, W. M. (2022). Intrusion detection: A model based on the improved vision transformer. Transactions on Emerging Telecommunications Technologies, 33(9), e4522. https://doi.org/10.1002/ett.4522

[9] Zhang, Z., & Wang, L. (2022, March). An Efficient Intrusion Detection Model Based on Convolutional Neural Network and Transformer. In 2021 Ninth International Conference on Advanced Cloud and Big Data (CBD) (pp. 248-254). IEEE. https://doi.org/10.1109/CBD54617.2021.00050

[10] Paulauskas, N., & Auskalnis, J. (2017, April). Analysis of data preprocessing influence on intrusion detection using NSL-KDD dataset. In 2017 open conference of electrical, electronic and information

sciences (eStream) (pp. 1-5). IEEE. https://doi.org/10.1109/eStream.2017.7950325

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053 c1c4a845aa-Paper.pdf

[12] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. https://doi.org/10.1109/CISDA.2009.5356528

[13] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE. https://doi.org/10.1109/10.1109/CISDA.2009.5356528