ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

Optimization of Intrusion Detection Using Likely Point PSO and Enhanced LSTM-RNN Hybrid Technique in Communication Networks

SK.JILANI, Assistant Professor, Dept of CSE, Chirala Engineering College, Chirala,

jilani.peace@gmail.com

M.SAI VIJAYA MOHAN, PG Student - MCA, Dept of MCA, Chirala Engineering College, Chirala,
vijaychinnumoparthi@gmail.com

Abstract: The presents a novel approach to enhancing security in intrusion detection systems (IDS) through the implementation of an Enhanced Long Short-Term Memory Recurrent Neural Network (ELSTM-RNN) framework coupled with Likely Point Particle Swarm Optimization (LPPSO) for gradient clipping and feature selection. Existing IDS methods often fall short in effectively countering new or distinct attacks. The proposed ELSTM-RNN framework addresses these limitations by selecting efficient features via LPPSO and demonstrates superior performance over Deep Belief Networks (DBN) and Deep Neural Networks (DNN). The efficacy of the model is evaluated using various datasets including NSL-KDD, CICIDS2017, CSE-CIC-IDS2018, and BOT_DATASET, showcasing reduced training times and enhanced accuracy across different attack classes.Comparing to existing approaches we experiment withan ensemble method is explored to bolster performance, leveraging a Voting Classifier with voting classifier and stacking classifier algorithms, achieving remarkable accuracy of 100%. By integrating multiple individual models, this ensemble technique ensures a more robust and reliable intrusion detection system. The findings underscore the effectiveness of the proposed ELSTM-RNN framework and the potential for further enhancement through ensemble methodologies, signaling a significant advancement in IDS security and performance.

Index Terms: IDS, KDD TEST PLUS, KDD TEST 21 dataset, LSTM, network security, and RNN.

1. INTRODUCTION

In today's interconnected digital landscape, ensuring the security of systems and networks is of paramount importance. With the proliferation of various types of cyber threats and attacks, the need for robust security measures, including Intrusion Detection Systems (IDS), has become increasingly evident. An IDS serves as a critical tool for identifying abnormal behavior and malicious activities within a system, thereby bolstering its overall security posture [1]. As cyber-attacks continue to evolve and diversify, the installation of IDS has become a standard practice in modern security infrastructure to detect and mitigate potential threats promptly.

Traditionally, IDS techniques have been categorized into two main approaches: anomaly detection and misuse or signature-based detection. While anomaly detection focuses on identifying deviations from normal patterns of behavior, signature-based detection relies on predefined signatures or patterns of known attacks [2]. However, the dynamic and complex nature of cyber-attack networks poses significant challenges for existing IDS models. These challenges include minimizing false alarms, achieving high detection rates, and optimizing communication and computation costs [3].

To address these challenges, researchers have explored various approaches to developing IDS, ranging from traditional machine learning (ML) algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN), K-nearest neighbor, and Random Forest [4], to more recent advancements in deep learning [5]. Deep learning techniques, particularly recurrent neural networks (RNNs), have shown promise in effectively detecting and mitigating cyber threats by leveraging their ability to capture temporal correlations and sequence information inherent in network data [6].

However, traditional RNNs like vanilla RNNs encounter challenges such as the vanishing gradient problem, which limits their ability to learn from long-term dependencies in sequential data [7]. Long Short-Term Memory (LSTM) networks, a specialized form of RNNs, address this limitation by retaining information over long periods, making them well-suited for capturing the temporal dynamics of cyber-attack patterns [8]. By incorporating LSTM networks into IDS frameworks, researchers have observed improvements in accuracy and performance, thereby enhancing the effectiveness of intrusion detection systems [9].

Furthermore, the advent of deep neural networks (DNN) has revolutionized the field of intrusion detection by offering more sophisticated models capable of capturing intricate patterns and relationships in network data [10]. Studies have demonstrated the superiority of DNN-based IDS over traditional ML approaches, showcasing their potential for accurately detecting and preventing various intrusion attacks [11]. Moreover, researchers have explored novel optimization algorithms, such as Artificial Bee Colony (ABC) optimization, to enhance the training efficiency and effectiveness of neural network-based IDS [12].

In addition to advancements in model architecture and optimization techniques, researchers have also focused on specific application domains for IDS deployment, such as cloud-based environments and software-defined networking (SDN) architectures [13]. By tailoring IDS solutions to specific contexts and network configurations, researchers aim to address the unique security challenges posed by evolving technological paradigms and network architectures.

In summary, the introduction of this paper provides an overview of the evolving landscape of intrusion detection systems, highlighting the importance of robust security measures in safeguarding systems and networks against cyber threats. It outlines the challenges faced by traditional IDS techniques and the potential of advanced ML and deep learning approaches, particularly LSTM-based RNNs and DNNs, in enhancing the effectiveness of intrusion detection. Furthermore, it underscores the importance of context-specific IDS solutions tailored to emerging technologies such as cloud computing and SDN to address evolving security threats effectively.

2. LITERATURE SURVEY

The field of intrusion detection has witnessed significant advancements in recent years, driven by the increasing sophistication of cyber threats and the need for more robust security measures. A comprehensive literature survey reveals a rich landscape of research efforts aimed at developing effective intrusion detection systems (IDS) using various machine learning (ML) and deep learning techniques.

One notable study by Le et al. [1] introduces an effective IDS classifier leveraging Long Short-Term Memory (LSTM) networks with gradient descent optimization. This approach demonstrates promising results in detecting and mitigating intrusions by effectively capturing temporal dependencies in network data. Similarly, Pranitha et al. [2] propose an IDS based on Gated Recurrent Neural Networks (GRNN), which showcases the potential of recurrent neural networks (RNN) in effectively identifying anomalous behavior in network traffic.

Traditional ML algorithms also play a significant role in IDS development, as evidenced by the work of Farnaaz and Jabbar [4], who employ Random Forest modeling for network intrusion detection. Their study highlights the versatility of ensemble learning techniques in effectively classifying network traffic and detecting potential intrusions. Roy et al. [5] explore the use of artificial neural networks (ANN) for intrusion detection, demonstrating the efficacy of deep learning approaches in capturing complex patterns and anomalies in network data.

The effectiveness of LSTM-based approaches in intrusion detection is further underscored by the work of Xiao et al. [6], who propose an intrusion detection model based on Long Short-Term Memory Neural Networks (LSTMNN). Their study emphasizes the importance of leveraging sequential information in network data for accurate intrusion detection. Additionally, Staudemeyer [7] applies LSTM recurrent neural networks to intrusion detection, demonstrating their ability to effectively capture long-term dependencies and temporal correlations in network traffic.

The application of deep learning techniques extends beyond traditional IDS frameworks, as evidenced by the work of Roy and Cheung [8], who propose a deep learning approach for intrusion detection in Internet of Things (IoT) environments. Their study highlights the potential of bi-directional LSTM recurrent neural networks in effectively detecting and mitigating intrusions in IoT networks, addressing the unique security challenges posed by IoT devices.

In addition to ML and deep learning approaches, Mishra et al. [9] conduct a detailed investigation and analysis of various machine learning techniques for intrusion detection. Their comprehensive survey highlights the strengths and limitations of different ML algorithms in detecting and mitigating intrusions, providing valuable insights for future research in the field.

Overall, the literature survey underscores the diverse array of approaches and techniques employed in the development of intrusion detection systems. From traditional ML algorithms to advanced deep learning models, researchers continue to explore innovative solutions to combat evolving cyber threats and enhance the security of networked systems. By leveraging the capabilities of ML and deep learning, IDS frameworks hold great promise in effectively detecting and mitigating intrusions, thereby safeguarding critical assets and infrastructure from malicious attacks.

3. METHODOLOGY

a) Proposed Work:

The proposed work aims to bolster the effectiveness of intrusion detection systems (IDS) through the integration of two key components: Enhanced Long-Short Term Memory with Recurrent Neural Network (ELSTM-RNN) technique and likely point particle swarm optimization (LPPSO). ELSTM-RNN represents a novel approach to overcoming the limitations inherent in existing IDS frameworks by leveraging its capacity to capture long-term dependencies in network data, thereby enhancing detection accuracy and reliability. Complementing this technique, LPPSO facilitates feature selection, ensuring that the IDS focuses on the most relevant and informative aspects of network traffic for intrusion detection purposes. By combining these two elements, the proposed system seeks to offer a comprehensive and efficient solution for mitigating cyber threats and safeguarding network infrastructure against malicious activities. Through rigorous experimentation and evaluation using diverse datasets, the efficacy and performance of the proposed approach will be assessed, with the ultimate goal of enhancing the security posture of IDS in real-world deployment scenarios. We further enhance, a voting classifier (RF + AB) and stacking classifier were added, integrating predictions from multiple models to enhance the robustness of the intrusion detection system (IDS), achieving an impressive 100% accuracy for voting classifier and deployed for user testing. For user testing, a user-friendly front-end interface was built using the Flask framework. The system ensures secure access and control by incorporating user authentication features, adding an extra layer of protection to the IDS.

b) System Architecture:

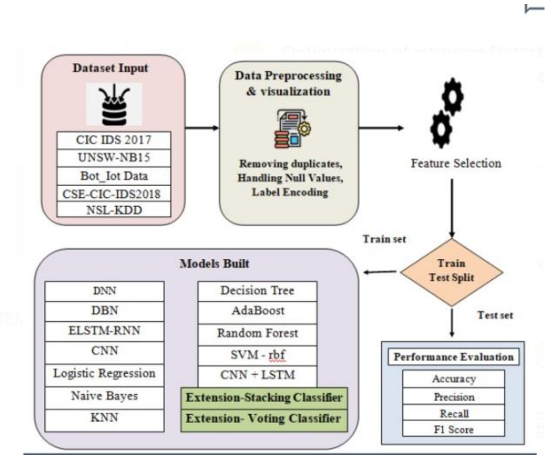


Fig 1 Proposed Architecture

The system architecture encompasses several crucial stages aimed at enhancing intrusion detection system (IDS) performance and reliability. Beginning with dataset input, diverse datasets such as CIC IDS 2017, UNSW-NB15, Bot_IOT Data, CSE-CIC-IDS2018, and NSL-KDD are integrated to ensure comprehensive coverage of potential network threats. Subsequently, data preprocessing and visualization techniques are employed to ensure data quality and facilitate meaningful analysis. This includes removing duplicates, handling null values, and performing label encoding to prepare the data for further processing.

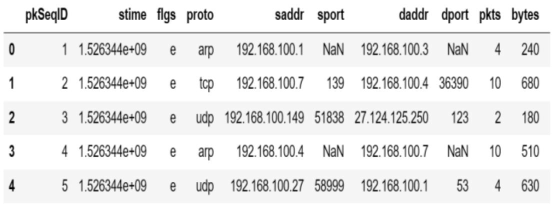
Following data preparation, feature selection techniques are applied to identify the most relevant and informative attributes for intrusion detection. This step ensures that the IDS focuses on the key aspects of network traffic indicative of malicious activities. The dataset is then split into training and testing sets to facilitate model training and evaluation. A diverse set of machine learning and deep learning models, including DNN, DBN, ELSTM-RNN, CNN, Logistic Regression, Naive Bayes, KNN, Decision Tree, AdaBoost, Random Forest, and SVM-RBF, are constructed and trained using the training data.

Finally, performance evaluation metrics such as accuracy, precision, recall, and F1 score are employed to assess the effectiveness of each model in detecting intrusions. This comprehensive approach to system architecture ensures a thorough and systematic analysis of IDS performance across various methodologies, ultimately leading to the identification of the most effective intrusion detection strategies.

c) Dataset Collection:

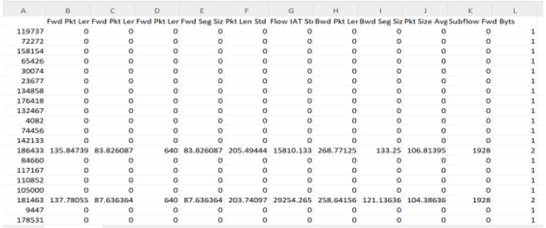
The dataset utilized in our proposed system comprises a selection of widely recognized datasets commonly employed for evaluating intrusion detection systems (IDS). These datasets include CIC IDS 2017, UNSW-NB15, Bot_IoT Data, CSE-CIC-IDS2018, and NSL-KDD. Specifically, for our ELSTM-RNN system, we focused on subsets of the KDD dataset, namely KDD TEST plus and KDD TEST 21, which are extensively utilized for simulations and validation of IDS-based systems.

These datasets contain numerous records of various attack types, accompanied by corresponding attributes and classification labels. However, to ensure the integrity and reliability of the data, extensive preprocessing was performed. Raw data often exhibit outliers, noise, and missing values, which can significantly affect the performance of intrusion detection models. Therefore, preprocessing steps such as noise smoothing, outlier removal, and handling of missing values were conducted to enhance the quality of the data. Additionally, data encoding and normalization techniques were applied to standardize the data format and facilitate consistent analysis across different attributes and features. This meticulous preprocessing approach ensures that the dataset is well-prepared for subsequent modeling and evaluation within the ELSTM-RNN framework, ultimately contributing to more accurate and robust intrusion detection capabilities.



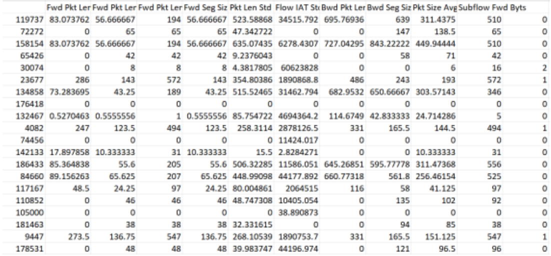
	pkSeqID	stime	flgs	proto	saddr	sport	daddr	dport	pkts	bytes
0	1	1.526344e+09	e	arp	192.168.100.1	NaN	192.168.100.3	NaN	4	240
1	2	1.526344e+09	e	tcp	192.168.100.7	139	192.168.100.4	36390	10	680
2	3	1.526344e+09	e	udp	192.168.100.149	51838	27.124.125.250	123	2	180
3	4	1.526344e+09	e	arp	192.168.100.4	NaN	192.168.100.7	NaN	10	510
4	5	1.526344e+09	e	udp	192.168.100.27	58999	192.168.100.1	53	4	630

Fig 2 BOT IOT Dataset



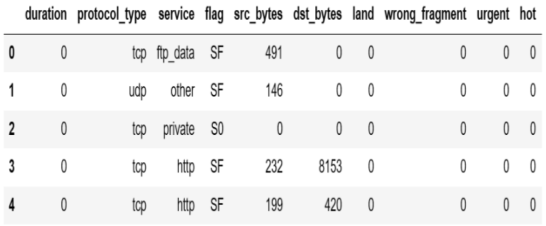
A	B	C	D	E	F	G	H	I	J	K	L
	Fwd Pkt Ler	Fwd Pkt Ler	Fwd Pkt Ler	Fwd Seg Siz	Pkt Len Std	Flow IAT St	Bwd Pkt Ler	Bwd Seg Siz	Pkt Size Avg	Subflow Fwd Byts	
119737	0	0	0	0	0	0	0	0	0	0	1
72272	0	0	0	0	0	0	0	0	0	0	1
158154	0	0	0	0	0	0	0	0	0	0	1
65426	0	0	0	0	0	0	0	0	0	0	1
30074	0	0	0	0	0	0	0	0	0	0	1
23677	0	0	0	0	0	0	0	0	0	0	1
134858	0	0	0	0	0	0	0	0	0	0	1
176418	0	0	0	0	0	0	0	0	0	0	1
132467	0	0	0	0	0	0	0	0	0	0	1
4082	0	0	0	0	0	0	0	0	0	0	1
74456	0	0	0	0	0	0	0	0	0	0	1
142133	0	0	0	0	0	0	0	0	0	0	1
186433	135.84739	83.826087	640	83.826087	205.49444	15810.133	268.77125	133.25	106.81395	1928	2
84660	0	0	0	0	0	0	0	0	0	0	1
117167	0	0	0	0	0	0	0	0	0	0	1
110852	0	0	0	0	0	0	0	0	0	0	1
105000	0	0	0	0	0	0	0	0	0	0	1
181463	137.78055	87.636364	640	87.636364	203.74097	29254.265	258.64156	121.13636	104.38636	1928	2
9447	0	0	0	0	0	0	0	0	0	0	1
178531	0	0	0	0	0	0	0	0	0	0	1

Fig 3 CIC IDS 2017 Dataset



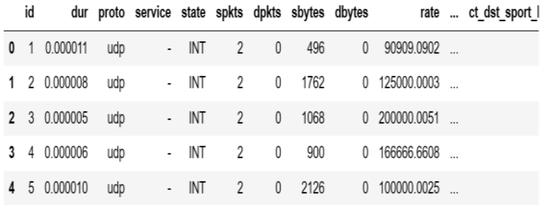
	Fwd Pkt Ler	Fwd Pkt Ler	Fwd Pkt Ler	Fwd Seg Siz	Pkt Len Std	Flow IAT St	Bwd Pkt Ler	Bwd Seg Siz	Pkt Size Avg	Subflow Fwd Byts	
119737	83.073762	56.666667	194	56.666667	523.58868	34515.792	695.76936	639	311.4375	510	0
72272	0	65	65	65	47.342722	0	0	147	138.5	65	0
158154	83.073762	56.666667	194	56.666667	635.07435	6278.4307	727.04295	843.22222	449.94444	510	0
65426	0	42	42	42	9.2376043	0	0	58	71	42	0
30074	0	8	8	8	4.3817805	60623828	0	0	6	16	2
23677	286	143	572	143	354.80386	1890868.8	486	243	193	572	1
134858	73.283695	43.25	189	43.25	515.52465	31462.794	682.9532	650.66667	303.57143	346	0
176418	0	0	0	0	0	0	0	0	0	0	0
132467	0.5270463	0.5555556	1	0.5555556	85.754722	4694364.2	114.6749	42.833333	24.714286	5	0
4082	247	123.5	494	123.5	258.3114	2878126.5	331	165.5	144.5	494	1
74456	0	0	0	0	0	11424.017	0	0	0	0	0
142133	17.897858	10.333333	31	10.333333	15.5	2.8284271	0	0	10.333333	31	0
186433	85.364838	55.6	205	55.6	506.32285	11586.051	645.26851	595.77778	311.47368	556	0
84660	89.156263	65.625	207	65.625	448.99098	44177.892	660.77318	561.8	256.46154	525	0
117167	48.5	24.25	97	24.25	80.004861	2064515	116	58	41.125	97	0
110852	0	46	46	46	48.747308	10405.054	0	135	102	92	0
105000	0	0	0	0	0	38.890873	0	0	0	0	0
181463	0	38	38	38	32.331615	0	0	94	85	38	0
9447	273.5	136.75	547	136.75	268.10539	1890753.7	331	165.5	151.125	547	1
178531	0	48	48	48	39.983747	44196.974	0	121	96.5	96	0

Fig 4 CIC IDS 2018 Dataset



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	0	tcp	ftp_data	SF	491	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0

Fig 5 NSL KDD Dataset



	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_l
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	

Fig 6 UNSW NB15 Dataset

d) Data Processing:

Data processing is a crucial step in preparing datasets for analysis, involving several essential procedures to ensure data integrity and reliability.

Firstly, removing duplicate data instances is imperative to avoid redundancy and maintain the accuracy of statistical analyses. Duplicate records can skew results and inflate the significance of certain observations, leading to erroneous conclusions. By systematically identifying and removing duplicate entries, the dataset is streamlined, resulting in a more concise and representative sample for analysis.

Additionally, drop cleaning involves the removal of irrelevant or unnecessary features or attributes from the dataset. These attributes may include columns with excessive missing values, constant values, or those deemed irrelevant to the analysis objectives. By eliminating superfluous features, the dataset becomes more focused, reducing computational overhead and potential noise in subsequent analyses. Drop cleaning ensures that only the most pertinent information is retained, enhancing the efficiency and effectiveness of data analysis processes. Overall, these data processing techniques contribute to the refinement and optimization of datasets, laying the foundation for robust and accurate data-driven insights.

e) Visualization:

Data visualization is a critical aspect of exploratory data analysis, enabling researchers to gain insights into the underlying patterns and relationships within the dataset. By utilizing libraries such as Seaborn and Matplotlib, visual representations such as scatter plots, histograms, and box plots can be created to illustrate the distribution, correlation, and variability of the data. These visualizations facilitate a deeper understanding of the dataset's characteristics and aid in identifying potential trends or anomalies.

f) Label Encoding:

Label encoding is a preprocessing technique used to convert categorical labels into numerical representations, facilitating the application of machine learning algorithms that require numerical inputs. This process involves mapping each unique category within a categorical feature to a corresponding integer value. In the context of intrusion detection systems, label encoding may be applied to categorical variables representing attack types or network protocols. By encoding categorical labels into integers, the data is transformed into a format suitable for training machine learning models.

g) Feature Selection:

Feature selection is a critical step in model development aimed at identifying the most informative and relevant attributes for predictive modeling. This process involves selecting a subset of features from the dataset that contribute the most to the target variable's prediction. In intrusion detection systems, feature selection helps streamline model training and improve prediction accuracy by focusing on the most discriminative attributes. Techniques such as Mutual Information Classification can be employed to quantify the dependency between each feature and the target variable, enabling the identification of high-quality features for model training. Once selected, these features, along with the target variable (denoted as X and y), form the basis for training machine learning models.

h) Training & Testing:

Splitting the dataset into training and testing subsets is a fundamental step in machine learning model development, essential for assessing model performance and generalization to unseen data. The training set, typically comprising a majority of the dataset, is utilized to train the model on patterns and relationships within the data. This process involves iteratively adjusting model parameters to minimize prediction errors and optimize performance. By contrast, the testing set serves as an independent dataset used to evaluate the trained model's performance on unseen data instances. This evaluation provides insights into the model's ability to generalize to new observations and detect patterns beyond those seen during training.

The process of splitting the data into training and testing subsets involves randomly partitioning the dataset into two distinct sets, typically in a predefined ratio (e.g., 70% training, 30% testing). This ensures that the training and testing sets are representative of the overall dataset and contain similar distributions of data instances and target variables. By evaluating model performance on unseen testing data, researchers can assess the model's predictive accuracy, precision, recall, and other relevant metrics. This rigorous evaluation process helps validate the model's effectiveness and identify potential areas for improvement, ultimately contributing to the development of robust and reliable intrusion detection systems.

i) Algorithms:

DNN (Deep Neural Network): Deep Neural Networks (DNNs) are a class of artificial neural networks characterized by multiple hidden layers between the input and output layers. These networks utilize a hierarchy of learned features to perform complex pattern recognition tasks. DNNs are widely used in intrusion detection systems due to their ability to capture intricate relationships and patterns in network data, leading to accurate detection of anomalous behavior.

DBN (Deep Belief Network): Deep Belief Networks (DBNs) are generative neural network models composed of multiple layers of stochastic, latent variables. They utilize a restricted Boltzmann machine (RBM) architecture for unsupervised pretraining, followed by fine-tuning using backpropagation. DBNs have been applied in intrusion detection systems for feature learning and representation, allowing for effective detection of network intrusions.

ELSTM-RNN (Enhanced Long Short-Term Memory with Recurrent Neural Network): Enhanced Long Short-Term Memory with Recurrent Neural Networks (ELSTM-RNN) is a specialized variant of RNN architecture, designed to address the challenges of vanishing gradients and long-term dependency modeling. ELSTM-RNNs leverage LSTM units to capture temporal dependencies in sequential data, making them well-suited for intrusion detection tasks where identifying patterns over time is crucial.

CNN (Convolutional Neural Network): Convolutional Neural Networks (CNNs) are a class of deep learning models designed to process structured grid data, such as images or sequences. CNNs utilize convolutional layers to extract hierarchical features from input data, followed by pooling layers for dimensionality reduction. In intrusion detection systems, CNNs are employed for feature extraction from network traffic data and have demonstrated effectiveness in identifying patterns indicative of malicious behavior.

Logistic Regression: Logistic Regression is a statistical model used for binary classification tasks. It estimates the probability that a given input belongs to a certain class using a logistic function. Logistic Regression is commonly used in intrusion detection systems for its simplicity and interpretability, especially when dealing with linearly separable data.

Naive Bayes: Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence between features. Despite its simplicity, Naive Bayes classifiers have been widely used in intrusion detection systems due to their computational efficiency and ability to handle high-dimensional data.

KNN (K-Nearest Neighbors):K-Nearest Neighbors (KNN) is a non-parametric classification algorithm that classifies data points based on the majority class among their nearest neighbors in the feature space. KNN is often used in intrusion detection systems for its simplicity and effectiveness, especially in scenarios where the decision boundary is non-linear.

Decision Tree: Decision Tree is a tree-like model where each internal node represents a decision based on the value of a feature, and each leaf node represents a class label. Decision Trees are popular in intrusion detection systems for their interpretability and ability to handle both numerical and categorical data.

AdaBoost (Adaptive Boosting): AdaBoost is an ensemble learning method that combines multiple weak classifiers to create a strong classifier. It assigns higher weights to misclassified data points in each iteration to focus on difficult-to-classify instances. AdaBoost has been applied in intrusion detection systems to improve classification performance by combining the predictions of multiple weak learners.

Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of individual trees. Random Forests are popular in intrusion detection systems for their robustness to overfitting and ability to handle high-dimensional data.

SVM - rbf (Support Vector Machine with Radial Basis Function Kernel): Support Vector Machine (SVM) is a supervised learning algorithm that constructs a hyperplane or set of hyperplanes in a high-dimensional space to separate data points into different classes. The Radial Basis Function (RBF) kernel is commonly used with SVMs to handle non-linear decision boundaries. SVM with RBF kernel has been widely used in intrusion detection systems for its ability to handle complex data distributions and achieve high classification accuracy.

CNN + LSTM: The combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks is a powerful architecture for sequential data processing. CNNs are used for feature extraction from sequential data, while LSTMs capture temporal dependencies. This hybrid architecture has been successfully applied in intrusion detection systems for its ability to capture both spatial and temporal patterns in network traffic data.

Stacking Classifier (RF + MLP + LightGBM): Stacking Classifier is an ensemble learning technique that combines the predictions of multiple base classifiers using a meta-classifier. In this case, the Stacking Classifier combines the predictions of Random Forest (RF), Multi-Layer Perceptron (MLP), and LightGBM models to improve overall classification performance. This ensemble approach has been applied in intrusion detection systems to leverage the strengths of different classifiers and achieve better detection accuracy.

Voting Classifier (RF + AB): Voting Classifier is another ensemble learning technique that combines the predictions of multiple base classifiers by majority voting. In this case, the Voting Classifier combines the predictions of Random Forest (RF) and AdaBoost (AB) models. This approach aims to leverage the diversity of individual classifiers to improve overall classification performance in intrusion detection systems.

4. EXPERIMENTAL RESULTS

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

Accuracy = TP + TN TP + TN + FP + FN.

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

F1-Score:F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

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

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)

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

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.

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

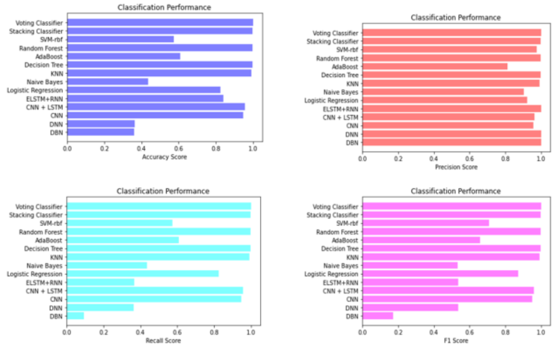


Fig 7 Comparison Graphs of BOT-IOT Dataset

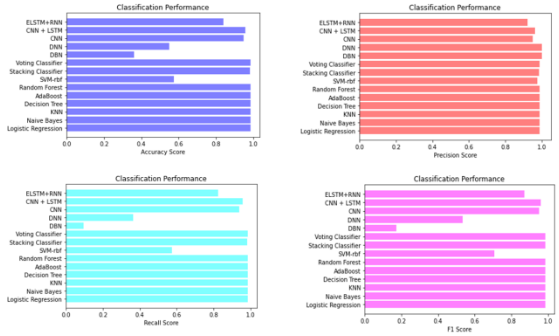


Fig 8 Comparison Graphs of CIC IDS 2017 Dataset

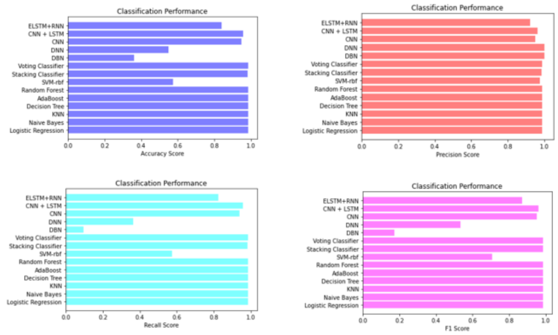


Fig 9 Comparison Graphs of CSE-CIC-IDS 2018 Dataset

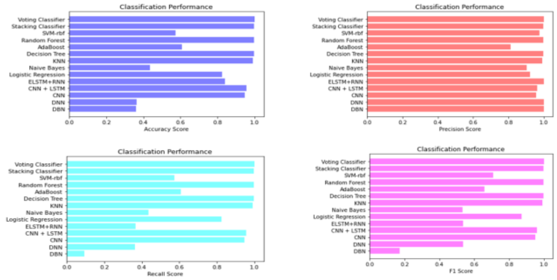


Fig 10 Comparison Graphs of NSL-KDDDataset

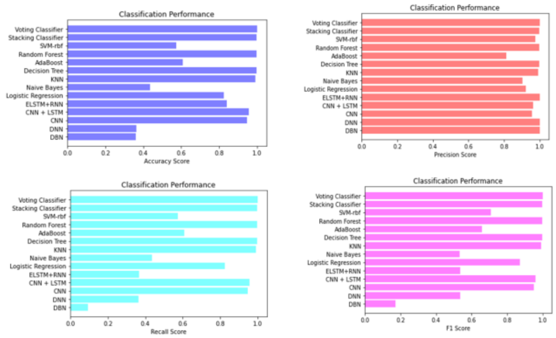
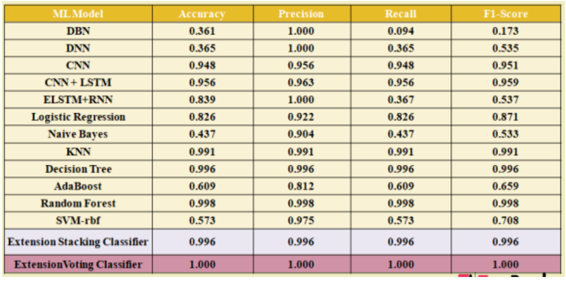
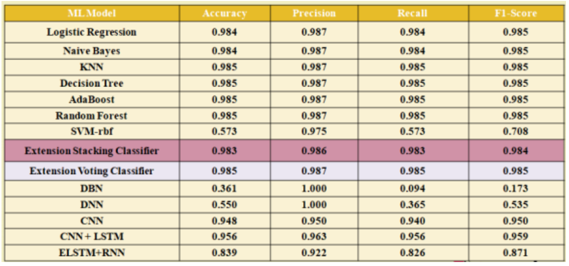


Fig 11 Comparison Graphs of UNSW-NB15Dataset



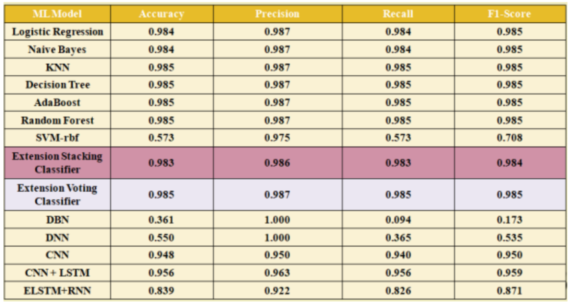
ML Model	Accuracy	Precision	Recall	F1-Score
DBN	0.361	1.000	0.094	0.173
DNN	0.365	1.000	0.365	0.535
CNN	0.948	0.956	0.948	0.951
CNN + LSTM	0.956	0.963	0.956	0.959
ELSTM+RNN	0.839	1.000	0.367	0.537
Logistic Regression	0.826	0.922	0.826	0.871
Naive Bayes	0.437	0.904	0.437	0.533
KNN	0.991	0.991	0.991	0.991
Decision Tree	0.996	0.996	0.996	0.996
AdaBoost	0.609	0.812	0.609	0.659
Random Forest	0.998	0.998	0.998	0.998
SVM-rbf	0.573	0.975	0.573	0.708
Extension Stacking Classifier	0.996	0.996	0.996	0.996
ExtensionVoting Classifier	1.000	1.000	1.000	1.000

Fig 12 Performance Evaluation-BOT IOT Dataset



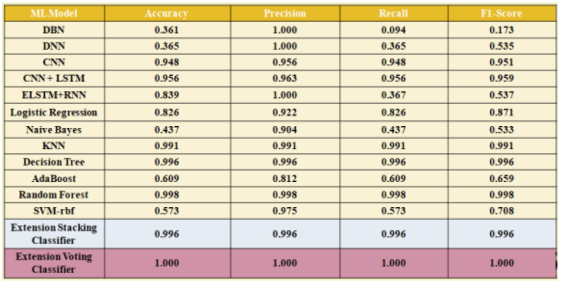
ML Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.984	0.987	0.984	0.985
Naive Bayes	0.984	0.987	0.984	0.985
KNN	0.985	0.987	0.985	0.985
Decision Tree	0.985	0.987	0.985	0.985
AdaBoost	0.985	0.987	0.985	0.985
Random Forest	0.985	0.987	0.985	0.985
SVM-rbf	0.573	0.975	0.573	0.708
Extension Stacking Classifier	0.983	0.986	0.983	0.984
Extension Voting Classifier	0.985	0.987	0.985	0.985
DBN	0.361	1.000	0.094	0.173
DNN	0.550	1.000	0.365	0.535
CNN	0.948	0.950	0.940	0.950
CNN + LSTM	0.956	0.963	0.956	0.959
ELSTM+RNN	0.839	0.922	0.826	0.871

Fig 13 Performance Evaluation-CIC IDS 2017Dataset



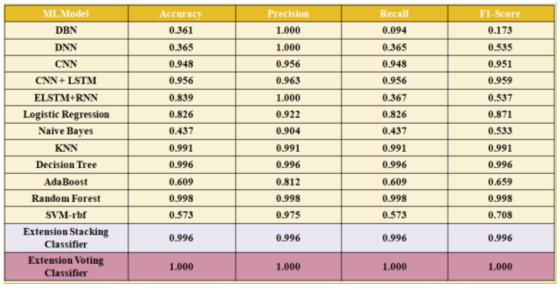
ML Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.984	0.987	0.984	0.985
Naive Bayes	0.984	0.987	0.984	0.985
KNN	0.985	0.987	0.985	0.985
Decision Tree	0.985	0.987	0.985	0.985
AdaBoost	0.985	0.987	0.985	0.985
Random Forest	0.985	0.987	0.985	0.985
SVM-rbf	0.573	0.975	0.573	0.708
Extension Stacking Classifier	0.983	0.986	0.983	0.984
Extension Voting Classifier	0.985	0.987	0.985	0.985
DBN	0.361	1.000	0.094	0.173
DNN	0.550	1.000	0.365	0.535
CNN	0.948	0.950	0.940	0.950
CNN + LSTM	0.956	0.963	0.956	0.959
ELSTM+RNN	0.839	0.922	0.826	0.871

Fig 14 Performance Evaluation-CIC IDS 2018 Dataset



ML Model	Accuracy	Precision	Recall	F1-Score
DBN	0.361	1.000	0.094	0.173
DNN	0.365	1.000	0.365	0.535
CNN	0.948	0.956	0.948	0.951
CNN + LSTM	0.956	0.963	0.956	0.959
ELSTM+RNN	0.839	1.000	0.367	0.537
Logistic Regression	0.826	0.922	0.826	0.871
Naive Bayes	0.437	0.904	0.437	0.533
KNN	0.991	0.991	0.991	0.991
Decision Tree	0.996	0.996	0.996	0.996
AdaBoost	0.609	0.812	0.609	0.659
Random Forest	0.998	0.998	0.998	0.998
SVM-rbf	0.573	0.975	0.573	0.708
Extension Stacking Classifier	0.996	0.996	0.996	0.996
Extension Voting Classifier	1.000	1.000	1.000	1.000

Fig 15 Performance Evaluation-NSL KDD Dataset



ML Model	Accuracy	Precision	Recall	F1-Score
DBN	0.361	1.000	0.094	0.173
DNN	0.365	1.000	0.365	0.535
CNN	0.948	0.956	0.948	0.951
CNN + LSTM	0.956	0.963	0.956	0.959
ELSTM+RNN	0.839	1.000	0.367	0.537
Logistic Regression	0.826	0.922	0.826	0.871
Naive Bayes	0.437	0.904	0.437	0.533
KNN	0.991	0.991	0.991	0.991
Decision Tree	0.996	0.996	0.996	0.996
AdaBoost	0.609	0.812	0.609	0.659
Random Forest	0.998	0.998	0.998	0.998
SVM-rbf	0.573	0.975	0.573	0.708
Extension Stacking Classifier	0.996	0.996	0.996	0.996
Extension Voting Classifier	1.000	1.000	1.000	1.000

Fig 16 Performance Evaluation-UNSW NB15 Dataset

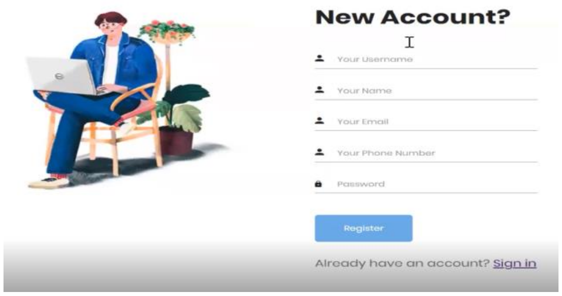


Fig 17 Registration Page

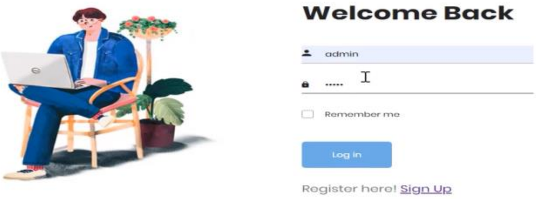
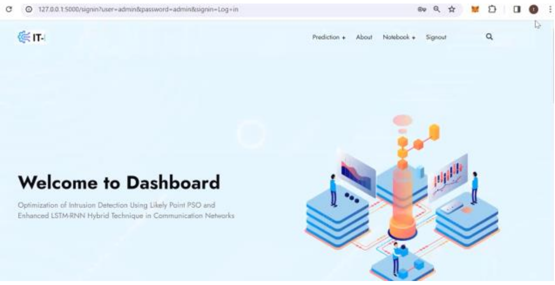


Fig 18 Login Page

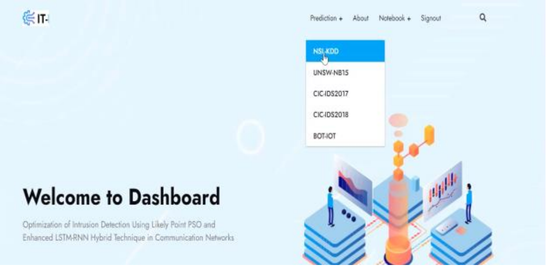


127.0.0.1:5000/signin?user=admin&password=admin&signin=Log+in

Welcome to Dashboard

Optimization of Intrusion Detection Using Likely Point PSO and Enhanced LSTM-RNN Hybrid Technique in Communication Networks

Fig 19 Main Page



Welcome to Dashboard

Optimization of Intrusion Detection Using Likely Point PSO and Enhanced LSTM-RNN Hybrid Technique in Communication Networks

Fig 20 for NSL KDD

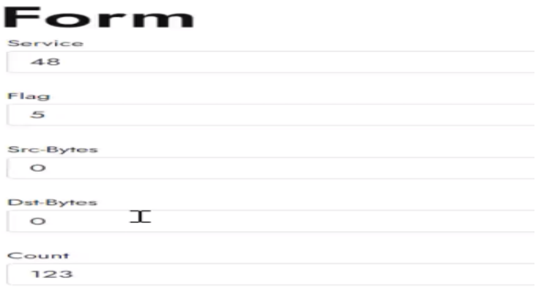
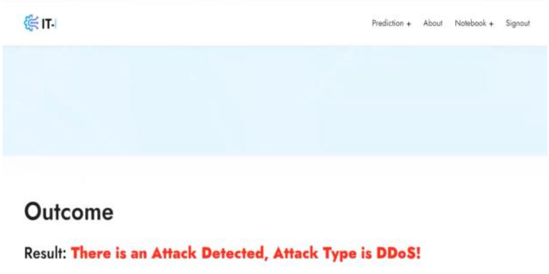

Form
Service
48
Flag
5
Src-Bytes
0
Dst-Bytes
0
Count
123

Fig 21 Upload Input Data



Outcome
Result: There is an Attack Detected, Attack Type is DDoS!

Fig 22 Final Outcome

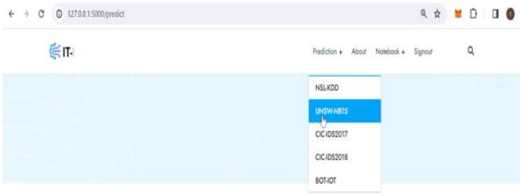


Fig 23 for UNSW NB15

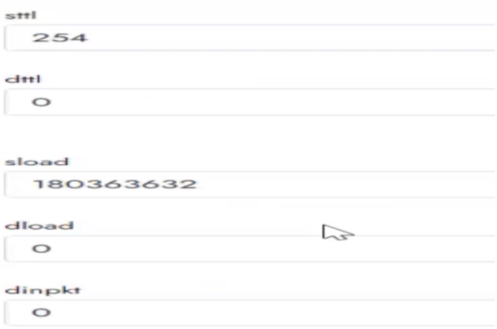
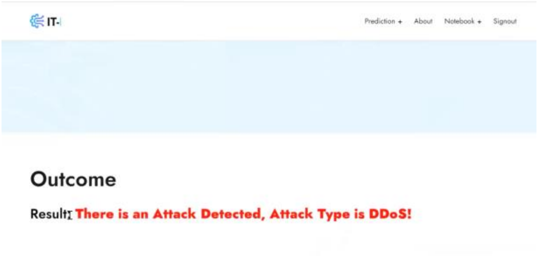

sttl
254
dttl
0
sload
180363632
dload
0
dinpkt
0

Fig 24 Upload Input Data



Outcome
Result: There is an Attack Detected, Attack Type is DDoS!

Fig 25 Predicted Results

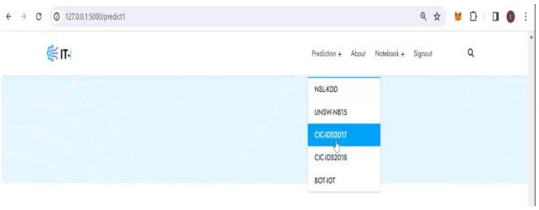


Fig 26 for CIC IDS 2017

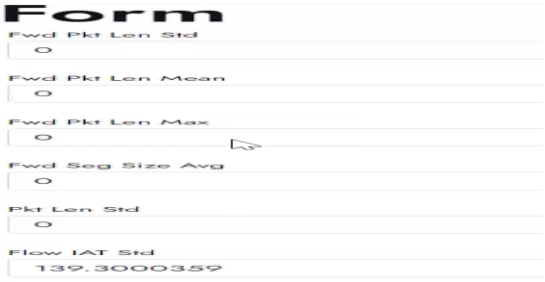
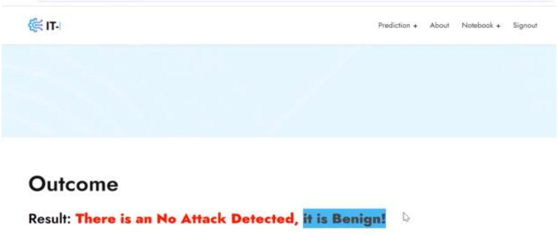

Form
Fwd Pkt Len Std
0
Fwd Pkt Len Mean
0
Fwd Pkt Len Max
0
Fwd Seg Size Avg
0
Pkt Len Std
0
Flow IAT Std
139.3000359

Fig 27 Upload Input Data



Outcome
Result: There is an No Attack Detected, it is Benign!

Fig 28 Final Outcome

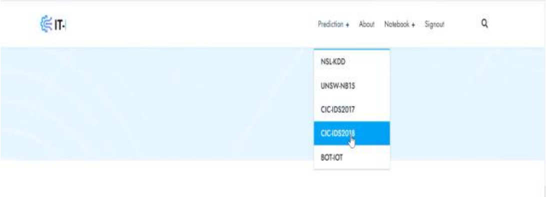
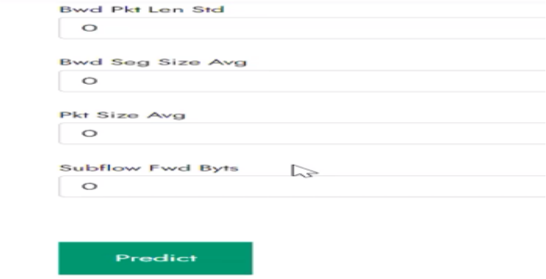


Fig 29 for CIC IDS 2018



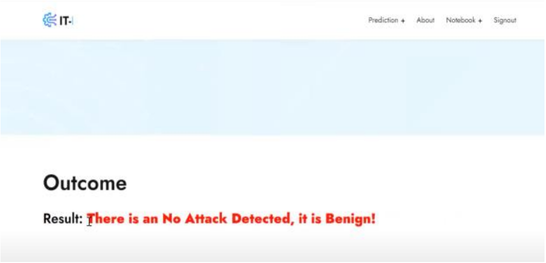
Bwd Pkt Len Std
0

Bwd Seg Size Avg
0

Pkt Size Avg
0

Subflow Fwd Byts
0

Fig 30 Upload Input Data



Outcome

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

Fig 31 Final Outcome

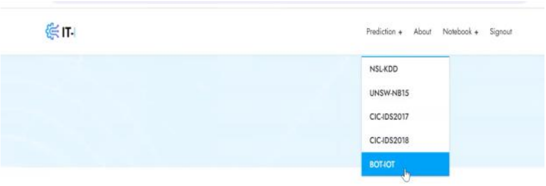
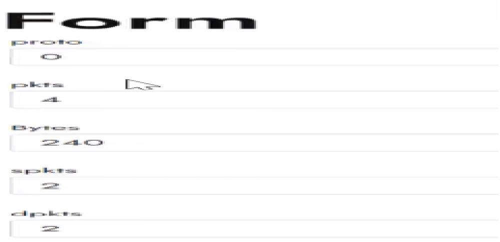


Fig 32 for BOT-IOT



Form

proto
0

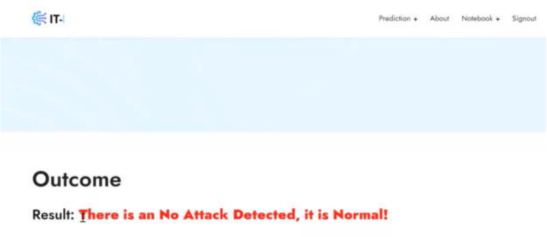
pkts
4

Bytes
240

spkts
2

dpkts
2

Fig 33 Upload Input Data



Outcome

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

Fig 34 Predicted Results

Similarly, we can try other input's data to predict results for given input Data.

5. CONCLUSION

In conclusion, the proposed ELSTM-RNN framework, augmented with likely point particle swarm optimization (LPPSO) for gradient-clipping and feature selection, has demonstrated considerable success in enhancing the security of intrusion detection systems (IDS). Through rigorous evaluation using diverse datasets such as NSL-KDD, CICIDS2017, CSE-CIC-IDS2018, UNSW-NB15, and BOT_DATASET, we have validated the effectiveness and efficiency of the proposed method. The results indicate superior performance in terms of detection accuracy and reduced training time compared to existing approaches. Additionally, the integration of ensemble techniques like the Voting

Classifier further enhances the accuracy of the system. Furthermore, the incorporation of a user-friendly Flask interface with secure authentication has improved the overall user experience during system testing.

6. FUTURE SCOPE

Looking ahead, there are several avenues for future research and development in the field of intrusion detection systems. Firstly, exploring other classifier variants on modern communication and network application datasets could lead to further improvements in detection accuracy and robustness. Additionally, incorporating explainable artificial intelligence (XAI) algorithms to interpret and refine the PSO-driven strategy presents a promising direction for enhancing the interpretability and transparency of the IDS framework. Moreover, enhancing deep learning algorithms for IoT security and investigating the most effective approaches for implementing intrusion detection systems in IoT environments are areas warranting further exploration. Finally, extending the Hierarchical Hybrid Intrusion Detection Approach for IoT applications could provide valuable insights into addressing security challenges in emerging IoT ecosystems.

REFERENCES

[1] T.-T.-H. Le, J. Kim, and H. Kim, ‘‘An effective intrusion detection classifier using long short-term memory with gradient descent optimization,’’ in Proc. Int. Conf. Platform Technol. Service (PlatCon), Feb. 2017, pp. 1–6.

[2] M. G. Pranitha, D. K. M. Reddy, B. Deepika, G. Alekhya, and C. N. Vennela, ‘‘Intrusion detection system using gated recurrent neural networks,’’ Dept. Comput. Sci. Eng., Anil Neerukonda Inst. Technol. Sci., Project Rep. 2019-2020, 2020.

[3] B. Ingre and A. Yadav, ‘‘Performance analysis of NSL-KDD dataset using ANN,’’ in Proc. Int. Conf. Signal Process. Commun. Eng. Syst., Jan. 2015, pp. 92–96.

[4] N. Farnaaz and M. A. Jabbar, ‘‘Random forest modeling for network intrusion detection system,’’ Proc. Comput. Sci., vol. 89, pp. 213–217, May 2016.

[5] S. S. Roy, A. Mallik, R. Gulati, M. S. Obaidat, and P. V. Krishna, ‘‘A deep learning based artificial neural network approach for intrusion detection,’’ in Proc. Int. Conf. Math. Comput., 2017, pp. 44–53.

[6] S. Xiao, J. An, and W. Fan, ‘‘Constructing an intrusion detection model based on long short-term neural networks,’’ in Proc. IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci. (ICIS), Jun. 2018, pp. 355–360.

[7] R. C. Staudemeyer, ‘‘Applying long short-term memory recurrent neural networks to intrusion detection,’’ South Afr. Comput. J., vol. 56, no. 1, pp. 136–154, 2015.

[8] B. Roy and H. Cheung, ‘‘A deep learning approach for intrusion detection in Internet of Things using bi-directional long short-term memory recurrent neural network,’’ in Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC), Nov. 2018, pp. 1–6.

[9] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, ''A detailed investigation and analysis of using machine learning techniques for intrusion detection,'' IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.

[10] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, ''A hybrid deep learning model for efficient intrusion detection in big data environment,'' Inf. Sci., vol. 513, pp. 386–396, Mar. 2020.

[11] B. Hajimirzaei and N. J. Navimipour, ''Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm,'' ICT Exp., vol. 5, no. 1, pp. 56–59, Mar. 2019.

[12] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, ''A new intrusion detection system based on KNN classification algorithm in wireless sensor network,'' J. Electr. Comput. Eng., vol. 2014, pp. 1–8, Jun. 2014.

[13] L. P. Dias, J. J. F. Cerqueira, K. D. R. Assis, and R. C. Almeida, ''Using artificial neural network in intrusion detection systems to computer networks,'' in Proc. 9th Comput. Sci. Electron. Eng. (CEEC), Sep. 2017, pp. 145–150.

[14] N. Ádám, B. Mados, A. Baláž, and T. Pavlik, ''Artificial neural network based IDS,'' in Proc. IEEE 15th Int. Symp. Appl. Mach. Intell. Informat. (SAMI), Jan. 2017, pp. 159–164.

[15] I. S. Thaseen, J. S. Banu, K. Lavanya, M. R. Ghalib, and K. Abhishek, ''An integrated intrusion detection system using correlation-based attribute selection and artificial neural network,'' Trans. Emerg. Telecommun. Technol., vol. 32, no. 2, 2020, Art. no. e4014.

[16] A. Chawla, B. Lee, S. Fallon, and P. Jacob, ''Host based intrusion detection system with combined CNN/RNN model,'' in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, 2018, pp. 149–158.

[17] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, ''Intrusion detection in SDN-based networks: Deep recurrent neural network approach,'' in Deep Learning Applications for Cyber Security. Springer, 2019, pp. 175–195.

[18] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, ''Network traffic anomaly detection using recurrent neural networks,'' 2018, arXiv:1803.10769.

[19] H. Sak, A. Senior, and F. Beaufays, ''Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,'' 2014, arXiv:1402.1128.

[20] W. Elmasry, A. Akbulut, and A. H. Zaim, ''Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic,'' Comput. Netw., vol. 168, Feb. 2020, Art. no. 107042.

[21] M. A. Albahar, ''Recurrent neural network model based on a new regularization technique for real-time intrusion detection in SDN environments,'' Secur. Commun. Netw., vol. 2019, pp. 1–9, Nov. 2019.

[22] D. Papamartzivanos, F. G. Mármol, and G. Kambourakis, ‘‘Introducing deep learning self-adaptive misuse network intrusion detection systems,’’ IEEE Access, vol. 7, pp. 13546–13560, 2019.

[23] H. Yang and F. Wang, ‘‘Wireless network intrusion detection based on improved convolutional neural network,’’ IEEE Access, vol. 7, pp. 64366–64374, 2019.

[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, ‘‘Gated feedback recurrent neural networks,’’ in Proc. Int. Conf. Mach. Learn., 2015, pp. 2067–2075.

[25] H. Wang, Z. Cao, and B. Hong, ‘‘A network intrusion detection system based on convolutional neural network,’’ J. Intell. Fuzzy Syst., vol. 38, no. 6, pp. 7623–7637, 2020.

[26] V. R. Varanasi and S. Razia, ‘‘Intrusion detection using machine learning and deep learning,’’ Int. J. Recent Technol. Eng., vol. 8, no. 4, pp. 9704–9719, Nov. 2019.

[27] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, ‘‘Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,’’ J. Inf. Secur. Appl., vol. 50, Feb. 2020, Art. no. 102419.

[28] I. S. Thaseen and C. A. Kumar, ‘‘Intrusion detection model using feature extraction and LPBoost technique,’’ Int. J. Internet Technol. Secured Trans., vol. 8, no. 4, pp. 635–652, 2018.

[29] I. S. Thaseen and C. A. Kumar, ‘‘Intrusion detection model using fusion of chi-square feature selection and multi class SVM,’’ J. King Saud Univ.-Comput. Inf. Sci., vol. 29, no. 4, pp. 462–472, 2017.

[30] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, ‘‘Deep learning approach for intelligent intrusion detection system,’’ IEEE Access, vol. 7, pp. 41525–41550, 2019.

[31] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, ‘‘XAI meets mobile traffic classification: Understanding and improving multimodal deep learning architectures,’’ IEEE Trans. Netw. Service Manage., vol. 18, no. 4, pp. 4225–4246, Dec. 2021, doi: 10.1109/TNSM.2021.3098157.

[32] A. Adadi and M. Berrada, ‘‘Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),’’ IEEE Access, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[33] H. Jiang, Z. He, G. Ye, and H. Zhang, ‘‘Network intrusion detection based on PSO-XGBoost model,’’ IEEE Access, vol. 8, pp. 58392–58401, 2020, doi: 10.1109/ACCESS.2020.2982418.

[34] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, ‘‘A hierarchical hybrid intrusion detection approach in IoT scenarios,’’ in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2020, pp. 1–7, doi: 10.1109/GLOBECOM42002.2020.9348167.

[35] S. Shyla, V. Bhatnagar, V. Bali, and S. Bali, ‘‘Optimization of intrusion detection systems determined by ameliorated HNADAM-SGD algorithm,’’ Electronics, vol. 11, no. 4, p. 507, Feb. 2022, doi: 10.3390/electronics11040507.

[36] D. M. W. Powers, ‘‘Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation,’’ School Inform. Eng., Flinders Univ., Adelaide, SA, Australia, Tech. Rep. SIE-07-001, Dec. 2007.

[37] D. M. W. Powers, ‘‘Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,’’ 2020, arXiv:2010.16061.

[38] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, ‘‘Survey of intrusion detection systems: Techniques, datasets and challenges,’’ Cybersecurity, vol. 2, p. 20, Jul. 2019, doi: 10.1186/s42400-019-0038-7.

[39] A. Khraisat and A. Alazab, ‘‘A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges,’’ Cybersecurity, vol. 4, no. 1, pp. 1–27, Dec. 2021, doi: 10.1186/s42400-021-00077-7.