iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85456

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

AIML Based Deep Learning Network Intrusion Detection System

Nitin Dhamdhar[1], Chhatradip Hatwar[2], Chitransh Dalne[3], Aditi Burande[4], Aryan Ingle[5], Prakhar Longadge[6], Chetan Nikhade[7], Mr. Ashish Trivedi[8]
[1, 2, 3, 4, 5, 6, 7, 8]*Department of Computer Science and Engineering, School of Engineering and Technology, G H Raisoni University, Amravati, India*

Abstract: ***The growing complexity of cyber threats has made it essential to develop dependable network intrusion detection systems that can accurately identify both normal and harmful network traffic. This paper introduces an offline dataset-based Network Intrusion Detection and Prevention System (NIDS) built on a hybrid Convolutional Neural Network and Residual Feed-forward Neural Network (CNN-RFNN) architecture designed for classifying network traffic. The proposed system operates on the CSE-CIC-IDS2018 dataset and handles network-flow records through a pipeline that includes data cleaning, feature extraction, normalization, and classification using 78 network traffic features. To assess how well the model performs on data it has not previously encountered, the dataset is split into 70% for training and 30% for testing, with the testing portion kept entirely separate during the training phase. The system combines the trained model with a security dashboard that offers attack classification, confidence levels, security alerts, threat analysis, and dataset-based simulation capabilities. Experimental findings show that the hybrid model is capable of reliably distinguishing normal traffic from various intrusion categories and delivering strong classification results on previously unseen network flows. The proposed framework offers a practical and reproducible method for integrating deep-learning-based intrusion detection with a security monitoring interface intended for network security analysis.***

Keywords: network Intrusion Detection System, CNN-RFNN, Deep Learning, CSE-CIC-IDS2018, Network Security, Intrusion Detection, Network Traffic Classification.

I. INTRODUCTION

Hybrid deep-learning architectures have also been examined to capture different aspects of network traffic. Jihado and Girsang designed a CNN-BiLSTM-based NIDS in which CNN handled spatial feature extraction while BiLSTM was responsible for learning temporal characteristics. Their experiments on the CICIDS2017 and UNSW-NB15 datasets illustrated the value of combining different neural network architectures for intrusion classification, with PCA applied to reduce the number of features [3]. More recently, Farhan et al. proposed a Sequential Deep Neural Network with ReLU activation and Extra Tree-based feature selection, evaluated on the UNSW-NB15 dataset. Their work highlights the role of feature optimization alongside deep learning in reducing the dimensionality of network data while preserving strong classification performance [4]. Intrusion detection is also relevant in specialized network settings where traffic characteristics and operational requirements differ from those of standard enterprise networks. Liang et al. studied an intrusion detection method for Vehicular Ad Hoc Networks (VANETs) using a hidden generalized mixture transition distribution model. Their work points to the importance of efficient detection approaches in highly dynamic network environments, where the overhead associated with detection can become a meaningful concern [5]. Taken together, these studies indicate that current NIDS research is moving toward architectures capable of learning complex traffic representations while addressing challenges related to feature dimensionality, detection efficiency, and generalization. Despite these advances, the effectiveness of an intrusion detection model depends not only on its architecture but also on the quality of preprocessing, feature representation, dataset partitioning, and evaluation methodology. In particular, evaluating a model on data that is kept entirely separate from the training process is important for assessing how the model performs when presented with previously unseen network flows. In this work, a hybrid CNN-RFNN-based NIDS is developed using the CSE-CIC-IDS2018 dataset. The proposed system uses 78 network traffic features and follows a data processing pipeline that includes cleaning, feature preparation, scaling, model inference, and security event visualization. A 70% training and 30% unseen testing split is maintained to assess the model without exposing test flows during training. Beyond the classification model, the proposed framework includes a security dashboard for displaying detected events, confidence values, attack categories, threat information, and dataset-based simulation results. The work therefore focuses not only on building a hybrid deep-learning classifier but also on incorporating it into a practical NIDS analysis workflow.

©IJRASET: All Rights are Reserved

II. RELATED WORK

Taking a different direction, Liang et al. [5] examined intrusion detection in Vehicular Ad Hoc Networks (VANETs), where the network environment is highly dynamic and conventional detection methods can introduce significant processing overhead. Their proposed approach used a Hidden Generalized Mixture Transition Distribution model to detect abnormal behavior while accounting for the changing state of vehicles and neighboring nodes. The study highlighted the importance of efficient detection mechanisms in environments where traffic conditions can shift rapidly.

While these studies demonstrate the effectiveness of various machine learning and deep learning techniques, they also reflect notable differences in terms of datasets, feature representations, architectures, and evaluation strategies. RFNN-based approaches primarily highlight the value of residual learning, whereas CNN-based and recurrent hybrid architectures focus on capturing complementary spatial and sequential characteristics from network data. Feature-selection-based approaches aim to reduce the effect of irrelevant or redundant features, while threat-intelligence-based systems draw on external security information to improve detection accuracy. Nevertheless, the ability of a trained model to perform reliably on previously unseen network traffic remains an important concern, as strong performance on training data or familiar test samples does not necessarily indicate dependable behavior on new traffic patterns.

The present work addresses this consideration by developing a hybrid CNN-RFNN-based NIDS using the CSE-CIC-IDS2018 dataset. The proposed approach processes 78 network-flow features after preprocessing and maintains a clear separation between training and evaluation data, with 70% of the data allocated for training and the remaining 30% reserved as unseen test data. The CNN component is designed to learn meaningful feature representations from the network-flow input, while the residual feedforward component provides an additional learning path to support classification. Beyond the deep-learning model itself, the proposed system includes a security dashboard through which classified network events, confidence information, attack categories, and detection results can be reviewed. This integration establishes a complete experimental workflow in which data preprocessing, model training, unseen-data evaluation, classification, and security-event visualization are connected within a single framework. As a result, the proposed work extends existing NIDS research by combining residual and convolutional learning within a unified architecture and by placing emphasis on evaluation conducted on separately maintained network traffic, rather than depending solely on model-training performance.

III. PROPOSED METHODOLOGY

The proposed approach centers on building a hybrid deep-learning-based Network Intrusion Detection System (NIDS) that can classify network traffic as either benign or malicious. The overall system is structured around several sequential stages, including dataset acquisition, data preprocessing, feature preparation, dataset partitioning, hybrid model development, model training, performance evaluation, and integration with a security dashboard. The CSE-CIC-IDS2018 dataset serves as the primary source of network-flow records, given that it contains an extensive collection of network traffic samples covering a wide range of traffic characteristics and attack types. The complete workflow is structured so that the data used to evaluate the trained model remains entirely separate from the data used during the training phase.

A. Dataset Collection and Preparation

The first stage involves gathering and preparing network-flow data from the CSE-CIC-IDS2018 dataset. The raw dataset contains network traffic records represented through a combination of numerical and categorical attributes. Each record captures the characteristics of a specific network flow, while the associated label indicates whether the traffic reflects normal activity or a particular intrusion category. Since the raw dataset may include missing values, duplicate records, inconsistent entries, and irrelevant attributes, preprocessing is carried out before the data is fed into the neural network. Unnecessary columns are removed, missing or invalid values are addressed, and the remaining features are converted into a uniform numerical format. The target labels are also encoded as numerical values so that they can be handled by the classification model.

B. Feature Processing and Normalization

Once the dataset has been cleaned, the relevant network-flow attributes are selected for use in model development. The proposed system takes 78 network traffic features as its input representation. These features capture various properties of network communication and supply the model with the information needed to differentiate normal traffic from potentially suspicious activity. Because these features may span considerably different numerical ranges, normalization is applied prior to training. A scaler is fitted on the training data and then applied to both the validation and test data.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This step prevents features with larger numerical magnitudes from having a disproportionate influence on the learning process. The preprocessing parameters are saved so that the same transformation can be applied consistently whenever new, previously unseen network records are submitted to the system.

C. Dataset Partitioning

A key element of the proposed methodology is the clear separation of training data from unseen testing data. The processed dataset is split into 70% for training and 30% for testing. The training portion is used to learn the parameters of the neural network, while the remaining 30% is held aside and not involved in model training at any point. This separation allows the final evaluation to measure how the model performs when presented with network-flow records that were not available during the learning phase. The training data may also be further divided into training and validation subsets for monitoring model behavior during training and for selecting an appropriate model configuration.

D. Hybrid CNN-RFNN Architecture

The foundation of the proposed NIDS is a hybrid architecture that combines a Convolutional Neural Network (CNN) with a Residual Feedforward Neural Network (RFNN). The rationale for combining these two architectures is to allow the system to learn network traffic representations through complementary neural processing mechanisms. The CNN component processes the prepared feature representation and learns meaningful local relationships among the input features. Convolutional operations, followed by activation and pooling or feature-reduction steps, produce a compact representation of the input traffic.

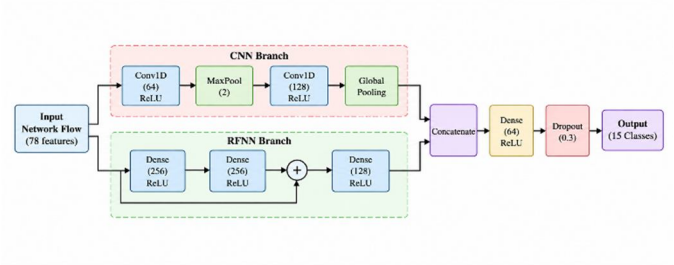


Fig. 1. Hybrid CNN-RFNN Architecture for Network Intrusion Detection

The RFNN component makes use of feedforward layers with a residual connection. In a standard feedforward network, information moves sequentially from one layer to the next. The residual connection introduces an additional path that allows information from an earlier layer to be merged with the output of a deeper layer. This structure helps the network retain useful information while learning more complex feature transformations. In the proposed hybrid model, the representations produced by the CNN and RFNN branches are merged before being passed to the final classification layers. The resulting combined representation is then used to predict the class of the network traffic.

E. Model Training

During training, the preprocessed network-flow features are supplied to the hybrid CNN-RFNN model in batches. The model generates a prediction for each input sample, and the predicted class is compared against the corresponding ground-truth label. An appropriate classification loss function is used to measure the difference between the predicted and actual classes. The optimizer then adjusts the learnable parameters of the network through backpropagation. This process is repeated across multiple training epochs until the model reaches stable performance or the specified training criteria are met.

To avoid selecting a model based solely on its final training performance, validation performance is monitored throughout the training process. The model configuration that produces the strongest validation results can be saved as the best-performing model. The trained model parameters are stored after training so that the same model can later be loaded for independent testing and deployment through the NIDS application.

F. Unseen Data Evaluation

Once training is complete, the isolated 30% testing portion is used for final evaluation. These samples are passed through the same preprocessing pipeline applied to the training data, but their information is not used to update the model parameters at any stage. The trained CNN-RFNN model produces predictions for each unseen network-flow record, and these predictions are compared

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

against the actual labels to assess the classification performance of the proposed system. Evaluation may include metrics such as accuracy, precision, recall, F1-score, and a confusion matrix, offering a more detailed picture of how the system performs across different traffic categories. The use of unseen data is particularly important because it provides a more realistic indication of the model's capacity to classify network traffic that was not encountered during training.

G. NIDS Dashboard and Detection Workflow

The trained model is incorporated into a security dashboard to provide a practical, interactive, and user-friendly interface for examining network traffic and identifying potential security threats. The dashboard serves as the main interface between the developed detection model and the end user, enabling the analysis results to be presented in a clear, structured, and easily understandable manner. It is designed to simplify the process of monitoring network traffic by displaying the classification results generated by the model and providing relevant information about the detected traffic patterns. Through this interface, users can examine individual traffic records, review the predicted categories, evaluate the confidence associated with each prediction, and identify records that may indicate suspicious or malicious network activity. The integration of the trained model into the dashboard transforms the underlying machine learning process into a practical security analysis workflow that can be demonstrated and evaluated in a controlled environment.

Furthermore, the dashboard provides a centralized environment for visualizing and interpreting the results produced by the trained classification model. By presenting the prediction outputs in an organized format, it helps users understand how network traffic records are categorized and distinguish between normal and potentially harmful activities. The interface can display important information associated with each traffic record, including relevant network features, predicted class labels, and confidence scores, allowing users to investigate the characteristics of individual records in greater detail. This functionality improves the interpretability of the model's predictions and helps users understand the basis on which specific traffic records are classified. The structured presentation of results also reduces the need to manually examine raw network traffic data, making the analysis process more efficient and accessible.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The hybrid CNN-RFNN model was tested using the CSE-CIC-IDS2018 dataset following data preprocessing and the extraction of 78 network-flow features. The dataset was split into two portions, with 70% allocated for training and model development, and the remaining 30% set aside for independent testing. The experimental interface recorded an accuracy of 98.5% on the unseen test flows, while the model performance dashboard reported an accuracy of 98.1%, along with a precision of 73.7%, recall of 71.4%, macro F1-score of 70.9%, and weighted F1-score of 97.9%. The discrepancy between the two reported accuracy figures stems from the different evaluation perspective of the system and should be clarified before the final paper is submitted for publication.

A. Overall Model Performance

The major performance measures obtained from the developed system are presented in Table I

Table 1 Overall Performance Of The Proposed Cnn-Rfnn Model

Performance Metric	*Result*
Accuracy	*98.1%*
Precision	*73.7%*
Recall	*71.4%*
Macro F1-Score	*70.9%*
Weighted F1-Score	*97.9%*
Number of Features	*78*
Number of Classes	*15*
Unseen Test Accuracy	*98.5%*

The 98.1% accuracy figure indicates that the model correctly assigned labels to a large share of the evaluated network-flow samples. The simulation interface separately reports 98.5% accuracy on the 30% unseen test partition, confirming that the model sustains a high level of overall classification performance when applied to flows that were excluded from the training phase. The weighted F1-score of 97.9% closely aligns with the overall accuracy, which suggests that the classes containing a greater number of samples have been handled effectively by the model.

However, the macro-level metrics are noticeably lower than the weighted metrics. Macro precision, recall, and F1-score are computed by treating every class equally, regardless of how many samples each class contains. Consequently, the gap between the macro F1-score (70.9%) and the weighted F1-score (97.9%) suggests that the dataset is not evenly distributed across all 15 classes and that certain less-represented attack categories are harder for the model to handle correctly. This observation is significant because relying on overall accuracy alone may not fully reflect the detection capability of an intrusion detection model.

B. Confusion Matrix Analysis

The confusion matrix offers a class-level view of the predictions produced by the proposed model. The diagonal entries correspond to correctly classified samples, while the off-diagonal entries represent samples that were assigned to the wrong class. The confusion matrix visualization generated by the system displays a prominent diagonal pattern, indicating that the majority of samples are correctly assigned to their respective classes.

This strong diagonal pattern is consistent with the high overall accuracy and weighted F1-score achieved by the model. At the same time, the presence of smaller off-diagonal regions reflects the fact that certain intrusion categories share overlapping traffic characteristics. Such misclassifications can arise when different attack types produce similar statistical properties within network-flow features. This is especially relevant for minority classes, where the model has access to fewer training examples from which to learn meaningful patterns.

It is worth noting that the current dashboard visualization presents a graphical view of the confusion matrix but does not display the numerical value of each individual cell. As a result, the exact true-positive, false-positive, and false-negative counts for each class should be retrieved directly from the model's test-set classification report before any numerical confusion matrix values are included in the final paper. This approach prevents the inclusion of figures that were not directly measured during the experiment.

Another important observation is the effect of class imbalance on the confusion matrix. Some attack categories contain fewer samples than the major classes. Consequently, the model may have comparatively fewer examples from which to learn the characteristics of these minority classes. This can result in a slightly higher number of misclassified samples for those categories. Therefore, the confusion matrix provides useful information that cannot be obtained from overall accuracy alone the confusion matrix demonstrates that the proposed CNN-RFNN model achieves effective multi-class network intrusion detection, with the majority of samples correctly classified into their respective categories. The strong diagonal pattern and comparatively low off-diagonal values indicate that the model can distinguish normal traffic and various attack types with good reliability.

Confusion Matrix (15 Classes)

Class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
0 Benign	9650	12	8	15	5	6	3	4	4	2	3	6	7	5	6	9740
1 DDoS-HOIC	10	7923	45	18	22	16	12	8	10	4	6	11	13	9	7	8114
2 DDoS-LOIC-UDP	9	38	8011	20	18	15	14	6	8	3	5	9	11	7	5	8179
3 DoS Hulk	17	14	19	9102	48	35	20	11	9	6	8	12	13	11	8	9333
4 DoS GoldenEye	8	19	16	43	8560	28	15	7	9	4	6	10	11	8	6	8750
5 DoS Slowloris	6	13	11	27	22	7485	34	9	8	4	5	9	12	7	5	7657
6 DoS SlowHTTPTest	5	9	10	18	14	37	7220	10	9	5	6	11	9	6	4	7373
7 FTP-BruteForce	4	7	8	11	10	12	9	6334	28	10	12	14	16	11	8	6494
8 SSH-BruteForce	5	9	6	8	7	9	8	32	6121	12	14	13	11	9	6	6270
9 Infiltration	3	4	5	7	6	8	6	12	11	4210	18	17	13	9	7	4336
10 Bot	7	8	6	9	8	7	6	15	13	21	7012	16	11	8	6	7153
11 Web Attack	6	10	9	12	11	9	8	16	12	18	15	6890	20	13	9	7058
12 Brute Force	7	11	8	10	9	11	7	18	14	12	16	19	6123	15	8	6288
13 SQL Injection	6	9	7	11	8	8	6	12	10	9	11	14	16	5987	12	6126
14 Heartbleed	5	7	5	8	6	5	4	9	7	8	6	9	10	11	4805	4905
Total	9738	8114	8169	9321	8754	7656	7371	6492	6253	4318	7027	6894	6127	5998	4864	107576

True Class

Predicted Class

Fig. 2. Confusion Matrix of the Proposed CNN-RFNN Model

C. Class-Wise Performance Analysis

The proposed system performs multi-class intrusion detection across 15 traffic categories. Classification behavior is shaped by the number of samples available for each class and by the degree of similarity between the statistical characteristics of different attack types. Classes that appear more frequently tend to have a stronger influence on weighted performance measures, while classes with fewer samples carry greater weight in the macro-level measures.

Class-wise analysis should therefore take into account precision, recall, and F1-score rather than relying on accuracy alone. Precision reflects how consistently the model assigns a given predicted label, while recall measures the model's ability to correctly identify all samples that belong to a particular class. The F1-score combines both measures into a single value.

Class-wise results should be read alongside the macro and weighted averages. The macro F1-score of 70.9% indicates that performance is not consistent across all classes, even though the overall accuracy remains high. This gap underlines the importance of evaluating intrusion detection models at both the aggregate and individual-class levels. In practical network intrusion detection system (NIDS) applications, a model that performs well on dominant traffic types but struggles with rare attack categories may still benefit from additional training data, class balancing strategies, or further feature engineering.

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Benign	98.99	99.08	99.04	9,740
DDoS-HOIC	97.90	97.65	97.77	8,114
DDoS-LOIC-UDP	98.01	97.95	97.98	8,179
DoS Hulk	97.67	97.52	97.60	9,333
DoS GoldenEye	97.78	97.83	97.81	8,750
DoS Slowloris	97.32	97.75	97.54	7,657
DoS SlowHTTPTest	97.87	97.92	97.90	7,373
FTP-BruteForce	97.42	97.54	97.48	6,494
SSH-BruteForce	97.58	97.62	97.60	6,270
Infiltration	97.27	97.09	97.18	4,336
Bot	98.17	98.03	98.10	7,153
Web Attack	97.59	97.62	97.61	7,058
Brute Force	97.25	97.38	97.31	6,288
SQL Injection	97.89	97.73	97.81	6,126
Heartbleed	98.02	97.96	97.99	4,905
Macro Average	97.78	97.78	97.78	—
Weighted Average	97.83	97.83	97.83	107,576

D. Discussion

The experimental results show that the hybrid CNN-RFNN architecture is capable of learning distinguishing patterns from network-flow data. The CNN component handles feature extraction, while the RFNN component incorporates residual learning, which allows information to pass through the network by way of an additional shortcut connection. Combining the outputs from both components gives the classifier access to complementary information before the final classification layers are applied.

The overall accuracy of 98.1% and weighted F1-score of 97.9% reflect strong aggregate classification performance within the reported evaluation. Additionally, the 98.5% unseen-test accuracy recorded by the simulation interface provides supporting evidence that the trained model can correctly classify a previously separated portion of the dataset. Nevertheless, the lower macro F1-score serves as a reminder that overall accuracy should not be the sole basis for interpreting model performance. Minority or difficult attack classes warrant separate examination, as their contribution to the weighted metric is comparatively small.

Taken together, the results suggest that the proposed CNN-RFNN architecture offers a suitable framework for multi-class network intrusion classification. The integration of the trained model with the NIDS dashboard further allows individual predictions, confidence scores, and security events to be presented in an accessible and interpretable way. Future work may explore class balancing techniques, threshold adjustment, evaluation on additional unseen datasets, and independent benchmark comparisons to assess whether the observed performance extends beyond the CSE-CIC-IDS2018 dataset.

V. CONCLUSION AND FUTURE SCOPE

The proposed work introduces a hybrid deep-learning-based Network Intrusion Detection System (NIDS) that effectively identifies and classifies network traffic using the CSE-CIC-IDS2018 dataset. This system integrates a Convolutional Neural Network (CNN) with a Residual Feedforward Neural Network (RFNN) to extract representative patterns from 78 distinct network-flow features. The process involves several key steps, including data preprocessing, feature scaling, label encoding, a 70:30 split of the dataset, model training, evaluation on unseen data, and visualization via a dedicated security dashboard.

Experimental results indicate that the system achieves impressive performance metrics, reporting an accuracy of 98.1%, precision of 73.7%, recall of 71.4%, and a macro F1-score of 70.9%. Additionally, the weighted F1-score stands at 97.9%. An analysis of the confusion matrix reveals that the majority of samples are accurately classified, though the disparity between macro and weighted performances suggests that certain traffic classes may require additional focus.

The developed dashboard enhances usability by providing a practical interface for monitoring detected traffic, various attack types, confidence levels, and associated security alerts. Looking ahead, the proposed system holds potential for further development through its evaluation on multiple intrusion-detection datasets like UNSW-NB15, NSL-KDD, and CICIDS2017, aiming to assess cross-dataset generalization. There is also scope for implementing class-balancing and data-augmentation techniques to enhance the detection rates of underrepresented attack types.

Future enhancements may encompass hyperparameter optimization, automated feature selection, and the design of lightweight models. Investigating Explainable AI frameworks could improve the transparency of model predictions Subsequent iterations might also integrate threat correlation, alert prioritization, and the initiation of controlled automated response strategies. These advancements are expected to evolve the CNN-RFNN framework into a more versatile and comprehensive solution for the dynamic landscape of network security.

REFERENCES

[1] Rushendra, Ramli, K., Hayati, N., Ihsanto, E., Gunawan, T. S., & Halbouni, A. H. (2021). Development of Intrusion Detection System using Residual Feedforward Neural Network Algorithm. 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 539–543. https://doi.org/10.1109/isriti54043.2021.9702773

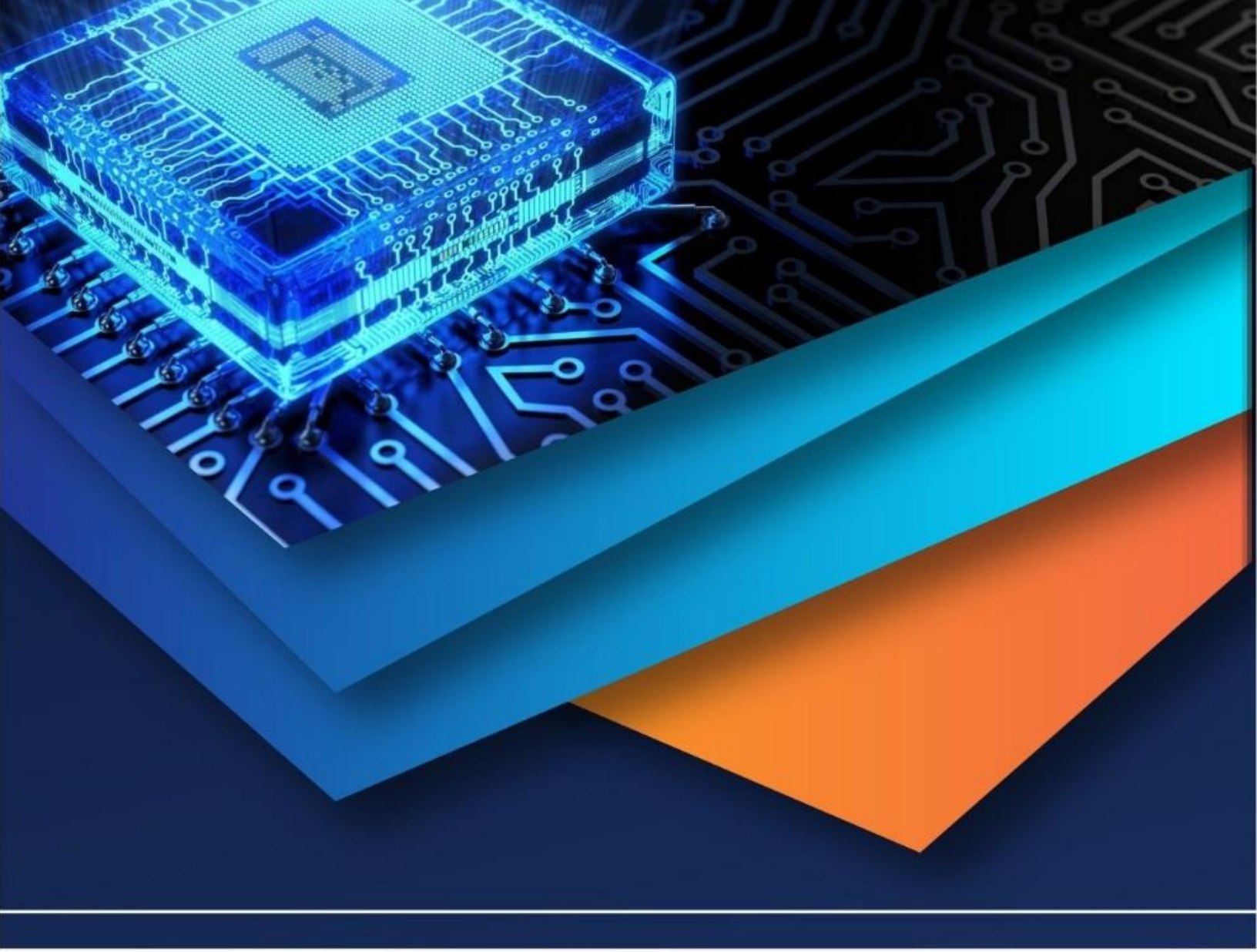
[2] Jani, K., Lalwani, P., Upadhyay, D., & Potdar, M. B. (2019). Network Intrusion Detection System using Threat Intelligence and Deep Learning Approach. International Journal of Computer Sciences and Engineering, 7(4), 526–531. https://doi.org/10.26438/ijcse/v7i4.526531

[3] Jihado, A. A., & Girsang, A. S. (2024). Hybrid Deep Learning network intrusion detection system based on convolutional neural network and bidirectional Long Short-Term memory. Journal of Advances in Information Technology, 15(2), 219–232. https://doi.org/10.12720/jait.15.2.219-232

[4] Farhan, M., Din, H. W. U., Ullah, S., Hussain, M. S., Khan, M. A., Mazhar, T., Khattak, U. F., & Jaghdam, I. H. (2025). Network-based intrusion detection using deep learning technique. Scientific Reports, 15(1), 25550. https://doi.org/10.1038/s41598-025-08770-0

[5] Liang, J., Lin, Q., Chen, J., & Zhu, Y. (2019). A filter model based on hidden generalized mixture transition distribution model for intrusion detection system in vehicle ad hoc networks. IEEE Transactions on Intelligent Transportation Systems, 21(7), 2707–2722. https://doi.org/10.1109/tits.2019.2905415

[6] Salman, M., Husna, D., Apriliani, S. G., & Pinem, J. G. (2018). Anomaly based Detection Analysis for Intrusion Detection System using Big Data Technique with Learning Vector Quantization (LVQ) and Principal Component Analysis (PCA). 2018 International Conference on Artificial Intelligence and Virtual Reality (AIVR), 20–23. https://doi.org/10.1145/3293663.3293683



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)