

Noise-Resilient Attention-Guided Deep Learning Framework for Network Intrusion Detection

Mrs. M. Sathya^{1*}, Dr. K. Pradeepa²

¹Ph.D. Research Scholar, School of Computing Science, KPR College of Arts, Science and Research, Coimbatore, India

Email: cmgsathya@gmail.com

²Professor and Dean, School of Intelligence Systems and Data Science, KPR College of Arts, Science and Research, Coimbatore, India

Email: drkpradeepa@gmail.com

*Corresponding Author: Mrs. M. Sathya, Ph.D. Research Scholar, School of Computing Science, KPR College of Arts, Science and Research, Coimbatore, India
Email: cmgsathya@gmail.com

ABSTRACT

The rising complexity of the state of the art network infrastructures and the swift advancement of cyber-attacks have rendered the correct and sound detection of intrusion to be a vital security need. Conventional intrusion detection solutions tend to collapse with noisy, high-dimensional network traffic, temporally changing traffic and encourage superior Deep Learning (DL)-based options. The main goal of this work is to improve the intrusion detection reliability by resolving the issue of noise sensitivity, over-representation of redundant features and inadequate temporal modelling. This is accomplished by introducing a noise-resilient attention-guided DL architecture which are Adaptive Noise-Resilient Data Preprocessing (ANR-DP), Hybrid Recursive Attention-Based Feature Selection (HRAFS), and a Long Short-Term Memory (LSTM)-based temporal classifier. ANR-DP acts dynamically on traffic noise and maintains the characteristics of the attacks, but HRAFS is expected to select stable and security-critical features through recursive elimination and attention-driven relevance learning. The optimized features are fed through an LSTM network in order to detect sequential and long-term intrusion patterns. Experimental analysis of the CICIDS2017 dataset provides a better performance with Proposed ANR-DP having low Missing Value Ratio of 1.2, High Outlier Reduction Rate of 91.5, HRAFS achieves a Feature Reduction Ratio of 68.7% and 9.1% accuracy improvement. The LSTM classifier also has an accuracy of 98.6%, and a low false positive rate of 3.1%. These findings establish that the proposed framework is very effective in enhancing robustness, precision, and dependability, it can be applicable in the environment of network intrusion detection.

Keywords: Attention Mechanism, Deep Learning, Intrusion Detection System, LSTM Networks, Network Security, Noise-Resilient Preprocessing

How to cite this article: Mrs M. Sathya, Dr K. Pradeepa. Noise-Resilient Attention-Guided Deep Learning Framework for Network Intrusion Detection. *Int J Drug Deliv Technol.* 2026;16(88s):250-257. DOI: 10.25258/ijddt.16.88s.25

.Source of support: Nil.

Conflict of interest: None

1. INTRODUCTION

The interconnected digital environments and network environments in the modern world have greatly expanded due to the rapid development of interconnected digital infrastructures, cloud services, Internet of Things (IoT) devices, and software-defined networks, which have made the environment quite more complex and large. This development has at the same time enhanced the frequency, types and complexity of cyber-attack, the making an assured intrusion detection of networks a very essential need in ensuring data confidentiality, integrity and service availability. Existing security systems are not able to handle intricate traffic patterns, high-dimensional data and periodical changing attack patterns, and therefore, smart, data-driven intrusion detection systems are embraced. Nevertheless, network traffic in the real world is always noisy, redundant and non-stationary, which negatively impacts detection accuracy and

constrains the practical implementation of most learning-based devices.

In recent studies, Machine Learning (ML) and Deep Learning (DL) methods have been applied to intrusion detection, such as multilayer neural networks [1], flow-based machine learning models on Software-Defined Networking (SDN) platforms [2], imbalance-sensitive deep learning models [3], hybrid Deep Convolutional Neural Network – Bidirectional Long Short-Term Memory (DCNN-BiLSTM) frameworks [4], and lightweight models on mobile IoT network devices [5]. IoT attack detection through supervised learning-based methods [6], Convolutional Neural Network (CNN) based deep intrusion detectors [7], hybrid CNN-RNN-based [8], and Recurrent Neural Network (RNN) driven temporal-based [9] have been shown to have better detection. Other researchers have concentrated on CNN-based Denial of Service (DoS) detection [10], deep learning in the detection of IoT intrusions [11], ensemble DL in detecting

anomalies [12], domain-specific Distributed Denial of Service (DDoS) detection [13], Convolutional Neural Network – Gated Recurrent Unit (CNN – GRU) hybrids [14], and optimum intrusion detection of fog and edge computing [15]. In spite of these improvements, there are still weaknesses in the adaptive noise handling, attention-directed feature prioritization, and stable temporal learning in the noisy traffic environment. In order to reduce them, this research propose a noise-resilient attention-guided model that combines adaptive preprocessing, hybrid recursive attention selection-based feature selection, and Long Short-Term Memory (LSTM) based temporal classifications. The first one is to enhance robustness, stability and detection reliability in real world network dynamics and the key contributions have been adaptive noise suppressions, attention based dimensionality reduction and ability to model sequential intrusion behavior.

Organization: Section 1 presents the research of the work aimed at creating a noise-resistant intrusion detection system. Section 2 describes the associated work and background research with the identification of the existing methods and the shortcomings. Section 3 summarizes the proposed methodology, namely adaptive preprocessing, attention-based feature selection, and LSTM-based classification, and Section 4 presents the findings of the experiment and further research solution, and Section 5 is the conclusion of the research with the most important findings and further research directions.

II. BACKGROUND STUDY

The article by L. Ashiku and C. Dagli (2021) [1] is dedicated to intrusion detection of network based on DL and integration of multilayer neural network to classify traffic. Minimal discussion of noisy data processing and no attention mechanisms show poor performance in cases of complicated attack distributions, although better performance in existing ML is achieved.

The article by A. O. Alzahrani and M. J. Alenazi (2021) [2] is devoted to the intrusion detection in software-defined networks, performed with the help of classical ML classifiers and flow-based features. Lacking deep feature learning and scalability test in large traffic scenarios is a weakness, but findings indicate the SDN can be trained with the level of accuracy.

Y. Fu et al. (2022) [3] recommend a DL model focused on the imbalanced intrusion data with the help of cost-sensitive learning and resampling techniques. Robustness against noise and attention-based methodology of feature selection are not addressed, but higher feature detection rates of minority classes of attacks are obtained.

V. Hnamte and J. Hussain (2023) [4] propose a DCNN-BiLSTM system architecture to model spatial and time-based attack patterns in the network

traffic. The restrictions include model complexity and lack of robustness examination in noisy or adversarial data, even at a high accuracy, and reduced false-positive levels.

The article by A. Amouri et al. (2020) [5] introduces a lightweight classifier-based intrusion detection system designed to address the needs of a mobile IoT setting based on ML. There are no deep representation learning and adaptive noise handling, but successful detection with low computation cost is shown.

Y. K. Saheed et al. (2022) [6] examines the application of the supervised machine learning models to IoT intrusion detection based on engineered features of the network. The inability to generalize between invisible attacks and lack of deep or attention learning are gaps, whereas experimental data prove that accuracy of classification is improved.

The article by A. Awajan (2023) [7] implements an intrusion detector of an IoT network using CNN architectures and based on DL. Weaknesses in the applicability are limited due to limited discussion of dataset imbalance and noise resilience, although high performance improvements over existing methods are also reported.

In the article M. A. Khan (2021) [8], a hybrid based on CNN-RNN is proposed to detect intrusions in the network traffic in real-time and sequentially. There are no attention mechanisms and noise tolerance to noisy traffic, even though the detection accuracy of various classes of attacks is enhanced.

S. M. Kasongo (2023) [9] proposes an intrusion detection framework based on the RNN framework and the focus on the temporal dependency of network traffic modeling. Scalability issues and inability to compare against attention-based deep models remain, and results have shown successful discovery of sequential patterns of attack.

TABLE I. DEEP LEARNING-BASED INTRUSION DETECTION SYSTEMS BACKGROUND STUDY

Authors (Year)	Concept	Methods Used	Research Gap	Limitations	Results
J. Kim et al. (2020) [10]	CNN-based intrusion detection for DoS attacks	CNN for traffic feature learning	Limited focus on only DoS attacks without multi-attack generalization	Performance degradation under diverse and noisy traffic patterns	Achieved high detection accuracy for DoS attacks with reduced false positives

B. Susilo & R. F. Sari (2020) [11]	IoT network intrusion detection	DL model applied to IoT traffic data	Lack of adaptive mechanisms for evolving IoT attack patterns	Scalability and real-time deployment constraints	Improved detection accuracy compared to existing ML methods
V. Dutta et al. (2020) [12]	Network anomaly and cyber-attack detection	DL ensemble models	Absence of attention mechanisms for feature prioritization	Increased computational complexity due to ensemble design	Demonstrated robust anomaly detection with enhanced overall accuracy
M. A. Ferrag et al. (2021) [13]	DDoS detection in Agriculture 4.0	DL-based IDS for distributed environments	Domain-specific design limits general applicability	Evaluation restricted to agriculture-focused datasets	Reported high DDoS detection accuracy in smart agriculture scenarios
B. Cao et al. (2022) [14]	Hybrid CNN – GRU intrusion detection model	CNN for spatial features and GRU for temporal patterns	No explicit noise-resilient or attention-guided learning strategy	Sensitivity to imbalanced datasets	Achieved improved accuracy and recall over standalone models
O. A. Alzubaidi et al. (2022)	Intrusion detection for fog and edge	Optimized machine learning models with	Limited exploration of deep attention-	Resource constraints in edge environment	Enhanced detection efficiency with reduce

[15]	computing	feature selection	based architectures	s	delay
------	-----------	-------------------	---------------------	---	-------

This table 1 introduces a background research of the recent DL-based intrusion detection methods, which identify the concepts, methodologies, research gaps, and limitations. The comparison shows that there are no attention-oriented and noise-tolerant mechanisms, which encourage advanced hybrid frameworks of strong network intrusion detection.

III. PROPOSED METHODOLOGY

This section introduces the noise-resilient attention-directed intrusion detection workflow, which describes the general algorithm design and end-to-end workflow. It outlines the architecture of the system, mathematical which rule adaptive preprocessing, feature selection, and temporal classification. Also, the section describes the full pseudocode to explicitly define the logical sequence and steps of implementation of the offered framework.

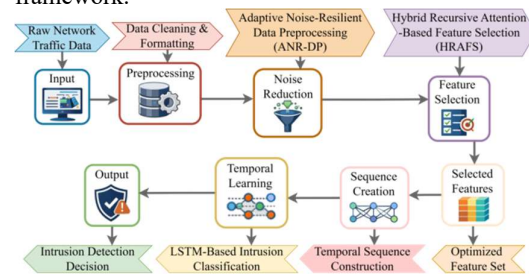


Fig. 1. The general architecture of the proposed ANR-DP-HRAFS-LSTM Intrusion Detection Framework.

This fig 1 shows the overall sequence of operations of the proposed intrusion detection system, with the beginning of acquisition and cleaning of raw network traffic, adaptive noise-resilient preprocessing, and hybrid recursive attention-based feature selection to extract an optimal set of features. The chosen features are arranged into time series and trained through an LSTM-based classifier to present a model that is efficient at capturing time-dependent patterns of attacks. Lastly, the framework will result in correct intrusion detection decisions through separation of normal traffic and different classes of attacks.

A. Dataset

Network Intrusion Detection dataset on Kaggle <https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection> is publicly available data set of network traffic records (which are labelled with normal and malicious) used to train and test intrusion detection systems. It has varied attack types and non-malicious traffic characteristics which makes it appropriate in machine learning based research in cyber security. The data can be used by the researchers to create and test models

that would effectively identify and categorize network intrusions under a real world condition.

B. Preprocessing using Adaptive Noise-Resilient Data Preprocessing

Existing approaches to statical normalization use fixed scaling or smoothing factors to all network traffic, and are inapplicable in practice when the nature of traffic is dynamically changing as a result of bursty flows, protocol heterogeneity and changing attack patterns. These non-adaptive methods either smooth the edges of important intrusion events or suppress random variations, resulting in unsteady distributions of features as well as compromised detection rates. To overcome this shortcoming, ANR-DP mechanism is proposed to adapt dynamically its preprocessing strength depending on the variability of the traffic and analysis of the inter-feature dependency. ANR-DP constantly assesses statistical dispersion and correlation between the features in order to detect those attributes that are dominated by noise and uses adaptive normalization and filtering respectively. The approach normalizes the feature representations under different traffic conditions by maintaining attack-relevant variations and eliminating unneeded variations. This learning behavior produces massive data consistency and learning reliability on downstream models. This step guarantees a strong and sound input in feature learning.

$$\eta_i = \frac{\sigma_i}{\mu_i + \epsilon}$$

(1)

Equation 1 is a representation of η_i is noise factor of the i^{th} feature, σ_i is the standard deviation of the i^{th} feature, μ_i is a mean value of the i^{th} feature and ϵ is a small constant to prevent instability of division. Relative noise intensity is quantified using this equation which compares the dispersion of features with the mean value of the feature.

$$x'_i = \frac{x_i - \mu_i}{\sigma_i \cdot (1 + \eta_i)} \quad (2)$$

Equation 2 denotes x_i is original feature value, x'_i are noise-adapted normalized feature, μ_i, σ_i means and standard deviation of that feature, η_i is adaptive noise factor. The higher noise levels are automatically adjusted to high strength such that noisy features would not prevail in learning.

$$S_i = 1 - \frac{\eta_i}{\max(\eta)}$$

(3)

Equation 3 indicates that S_i is stability weight of the i^{th} feature, η_i is noise factor of the feature, and $\max(\eta)$ is maximum noise factor of all features. The more noisy features are, the less stability is given to them, so that the same features are not significantly affected when training.

C. Feature Selection using Hybrid Recursive Attention-Based Feature Selection

The data of a high dimensional network traffic usually include redundant and weakly informative

feature that adversely impacts the learning effectiveness and the generalization of detection. Recursive Feature Elimination (RFE) methodically solves this problem by training a learner and removing features that contribute the least and then continues to further narrow the set of features. Nevertheless, RFE is not able to fit the intricate non-linear relationships between features in a deep learning context. To counter this weakness, HRAFS incorporates an attention mechanism that displays dynamic weights of importance on each feature in accordance with the contribution of each feature on intrusion discrimination. Attention module acquires local dependencies and global patterns of relevance such that the framework re-welcomes feature importance in an adaptive manner as the recursion process continues. HRAFS is an effective dimensionality reduction based on the joint action of recursive elimination and attention-guided weighting because feature stability is ensured across training sessions. HRAFS picks security-critical attributes and gets rid of redundancy.

$$F^{(0)} = \{f_1, f_2, \dots, f_n\}$$

(4)

Equation 4 denotes $F^{(0)}$ initial feature set, prior to the start of the selection, $f_i = i^{\text{th}}$ extracted network traffic feature, n total feature extracted. This equation is used to determine the entire set of extracted features that is the initial point of recursive feature selection.

$$e_i = w^T \tanh(W f_i + b)$$

(5)

Equation 5 is that e_i is raw attention score of feature f_i is input feature vector, W is learnable weight matrix, w is learnable attention vector, b is bias term, $\tanh$ is nonlinear activation function. This equation calculates a nonlinear relevance of every feature based on learnable attention parameters.

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^n \exp(e_j)}$$

(6)

Equation 6 is an indicator that α_i is a normalized attention score of feature f_i , e_i is raw attention score of feature f_i , and n is the total number of features. It is an equation that normalizes attention scores and takes the form of softmax to indicate the relative importance of each feature.

$$\tilde{f}_i = \alpha_i \cdot f_i$$

(7)

Equation 7 is $\tilde{f}_i$ is attention-weighted feature value, α_i is attention weight of f_i is original feature value. This equation takes into consideration the important features and leaves the less important features suppressed with the help of the weights of attention.

$$I_i = \frac{1}{T} \sum_{t=1}^T |\tilde{f}_i(t)|$$

(8)

Equation 8 indicates I_i is the importance score of feature f_i and $\tilde{f}_i$ is the attention-weighted feature at

training iteration t and T is the total number of training iterations. This equation obtains the feature importance that is resistant on the average of many training cycles.

$$F^{(k+1)} = F^{(k)} \setminus \{f_i \mid I_i < \tau\} \quad (9)$$

Equation 9 $F^{(k)}$ is feature set in iteration k , $F^{(k+1)}$ is feature set following elimination, I_i is the importance score of feature f_i , τ is feature elimination threshold. This is an equation which recursively eliminates features whose significance is less than a predetermined threshold.

$$F^* = \arg \max_F \sum_{f_i \in F} \alpha_i \quad (10)$$

In equation 10, F^* is a final optimized feature subset, F is candidate feature subset, α_i is attention weight on feature f_i . This equation chooses the best and most stable set of features by maximizing cumulative weights of attention.

D. Classification using LSTM-Based Temporal Intrusion Classification

Multi-stage attacks, slow-rate scanning and persistent threats are some of the sequential dependencies of the modern network intrusions that tend to evolve with time and cannot be easily detected with the help of the static classifiers. Therefore, temporal modeling is needed to store correlations between successive flows of traffic and learn patterns of attack progression. The LSTM networks are specifically applicable to the present task because the architecture is gated, and the can either retain or forget the information about time in long sequences selectively. The LSTM can learn short-term anomalies as well as long-range behavioral patterns by learning the ordered network traffic characteristics on a temporal basis. It operates under the dynamic updating of the internal states of the memory cells and gating mechanisms according to the changing traffic properties, through which the model is capable of distinguishing between benign and malicious temporal variations. This sequential learning ability enhances immensely in intrusion detection credibility in challenging network circumstances. This will help in the correct identification of both short-term and long-term attacks.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (11)$$

Equation 11 does i_t is input gate activation at time step t , x_t is input feature at time t , h_{t-1} is the previous hidden state, W_i is input feature weight matrix, U_i is previous hidden state weight matrix, b_i is input gate bias, and σ is the sigmoid activation function. This equation determines the amount of new network traffic information that is permitted to enter the LSTM memory cell.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (12)$$

Equation 12 is the f_t is forget gate activation vector, x_t is an input feature at time t , h_{t-1} is the previous hidden state, W_f is an input feature weight

matrix, U_f is a hidden state weight matrix, b_f is a forget gate bias term. This equation is used to decide which historical network traffic patterns are to be stored or forgotten.

$$C_t = f_t \times C_{t-1} + i_t \times \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (13)$$

Equation 13 is C_t is updated cell state at time t , C_{t-1} is a previous cell state, f_t is forget gate output, i_t is input gate output, W_c is weight matrix of candidate cell state, U_c weight matrix of contribution by hidden state, b_c cell state bias term, $\times$ is element-wise multiplication, $\tanh$ is a hyperbolic tangent activation. The equation is used to refresh the long-term memory by combining the retained past information with the new patterns of the traffic that is learned.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad h_t = o_t \times \tanh(C_t) \quad (14)$$

Equation 14 is the expression of o_t is output gate activation vector, h_t is current hidden state, x_t is input feature vector, h_{t-1} is previous hidden state, W_o is output gate weight matrix, U_o is hidden state weight matrix, b_o is output gate bias term. The given equation produces a time-filtered hidden representation, which contains significant patterns in intrusion.

$$\hat{y}_t = \text{Softmax}(W_y h_t + b_y) \quad (15)$$

Equation 15 denotes $\hat{y}_t$ is predicted intrusion class probability vector, h_t is secret state at period t , W_y is output layer weighting matrix, b_y is output bias term and Softmax is probability normalization function. This equation projects learned temporal representations to end intrusion detection class probabilities.

Algorithm: Noise-Resilient Attention-Guided LSTM-Based Intrusion Detection Framework

```

Begin
Input raw network traffic dataset D
Handle missing values and inconsistent entries in D
Initialize preprocessed dataset D_pre
// Stage 1: Adaptive Noise-Resilient Data
Preprocessing (ANR-DP)
For each feature  $f_i$  in D do
    Analyze traffic variability and statistical
    dispersion of  $f_i$ 
    Apply adaptive normalization and noise
    filtering to  $f_i$ 
    Store noise-resilient feature in D_pre
End For
// Stage 2: Hybrid Recursive Attention-Based
Feature Selection (HRAFS)
Initialize feature set F from D_pre
Repeat
    Train attention module using current feature set
    F
    Compute attention-based importance scores for
    all features

```

```

    Identify low-importance and redundant
    features
    Remove least significant features from F
    Until feature importance convergence or predefined
    feature count reached
    Store optimized feature subset F_opt
    // Stage 3: LSTM-Based Temporal Intrusion
    Classification
    Construct temporal sequences from F_opt
    Initialize LSTM network parameters
    Train LSTM model using sequential feature data
    For each test sequence do
        Predict intrusion class using trained LSTM
    End For
    Output predicted intrusion labels Y
End

```

The proposed pseudocode details the process of three phases of an intrusion detection pipeline that takes raw network traffic and the results are used to make accurate predictions of intrusion. First, ANR-DP optimizes the traffic features by suppressing noise dynamically, without losing information that will be used in the attack. Then, HRAFS recursively attaches importance to features and eliminates the redundant feature to get an optimized feature set. Lastly, a sequence of temporal features is categorized based on an LSTM network, which allows the detection of not only short-term anomalies but also long-lasting intrusion patterns as well.

IV. RESULT AND DISCUSSION

This section is a detailed analysis of the given intrusion detection framework where all the preprocessings, feature selection, and classification experiments were performed and analyzed in Python based simulation and analysis tools. Systematic reporting of the results is done using quantitative measures to determine the ability to reduce noise, feature relevance, and classification. The existing DL models are compared to make the improvements in the performance at each step of the framework noticeable. The results of this experiment show that the offered methods achieve higher results than existing ones in all the metrics considered.

TABLE II. COMPARISON OF ANR-DP PREPROCESSING PERFORMANCE METRICS

Method	MVR (%)	ORR (%)	FVS	SNR (dB)
DNN [1]	4.6	72.3	0.81	14.2
Autoencoder [3]	3.9	75.6	0.84	15.1
CNN [10]	3.1	78.9	0.87	16.4
CNN-GRU [14]	2.4	82.7	0.90	17.8
Proposed ANR-DP	1.2	91.5	0.95	21.3

Table II compares the effectiveness of the DNN, Autoencoder, CNN, CNN-GRU, and the proposed ANR-DP in preprocessing in terms of Missing

Value Ratio (MVR), Outlier Reduction Rate (ORR), Feature Variance Stability (FVS), and Signal-to-Noise Ratio (SNR). The proposed ANR-DP has the lowest MVR and the largest ORR, FVS, and SNR, which means that it has better noise suppression and feature stability to ensure accurate intrusion detection.

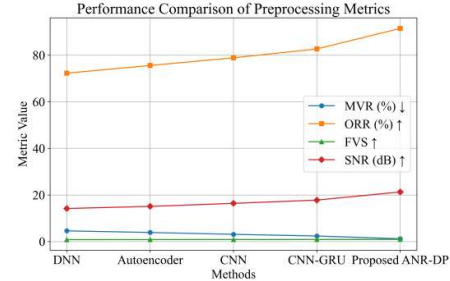


Fig.2. Comparison of Metrics of Preprocessing Performance

This fig 2 shows the comparison of preprocessing techniques with x-axis showing various techniques (DNN, Autoencoder, CNN, CNN-GRU and Proposed ANR-DP) and y-axis showing the values of the metrics of MVR (%), ORR (%), FVS, and SNR (dB). Proposed ANR-DP method boasts lower rate of missing value as well as outlier-fit removal, validity of features and signal to noise ratio; as opposed to other techniques.

TABLE III. PERFORMANCE OF THE FEATURE SELECTION ON HRAFS.

Method	FRR (%)	MI	Δ Accuracy (%)	SS
DNN [1]	34.6	0.62	2.8	0.71
Autoencoder [3]	41.2	0.67	3.6	0.75
CNN [10]	48.9	0.72	4.9	0.81
CNN-GRU [14]	56.4	0.78	6.3	0.86
Proposed HRAFS	68.7	0.85	9.1	0.93

Table III compares the performance of feature selection of DNN, Autoencoder, CNN, CNN-GRU and the proposed HRAFS on Feature Reduction Ratio (FRR), Mutual Information (MI), Accuracy Improvement (DeltaAccuracy), and Stability Score (SS). The HRAFS proposed provides the best FRR, MI, ΔAccuracy, and SS, and it is effective in reducing its dimensions and maintaining the discriminative and stable security-relevant features.

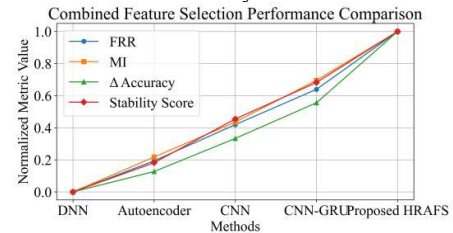


Fig. 3. Performance Comparison of the combined feature selection

This fig 3 is a comparative analysis of the feature selection technique, with the x-axis showing the various techniques (DNN, Autoencoder, CNN, CNN-GRU and Proposed HRAFS) and the y-axis showing the normalized value of feature selection technique in the terms of the FRR, MI, 0.09 Accuracy and Stability Score. The Proposed HRAFS method also has significantly higher normalized scores, indicating better feature relevance, stability as well as accuracy improvement than existing methods.

TABLE IV. COMPARISON OF THE CLASSIFICATION PERFORMANCE OF LSTM

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FP R (%)
DNN [1]	88.4	87.6	86.9	87.2	6.8
BiLSTM [4]	90.1	89.4	88.8	89.1	5.9
RNN [9]	90.2	88.5	89.7	87.1	7.4
CNN [10]	95.3	93.7	94.1	92.4	5.2
Proposed LSTM	98.6	97.0	97.5	96.7	3.1

Table IV compares the classification of DNN, BiLSTM, RNN, CNN and the proposed model of LSTM on the basis of Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). It is demonstrated that the proposed LSTM has the highest detection accuracy and F1-score and the lowest FPR, which proves its utility in predicting time-related intrusion trends.

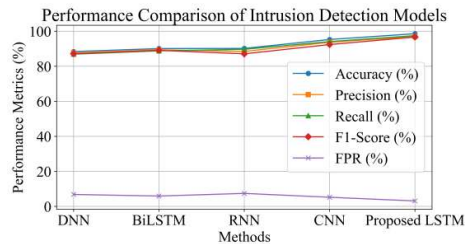


Fig. 4. Compared Results of the Intrusion Detection Models

This fig 4 shows the comparative results of various intrusion detection approaches whereby the x-axis is the models of classification (DNN, BiLSTM, RNN, CNN, and Proposed LSTM) whereas the y-axis is the performance measures in the form of (%) of Accuracy, Precision, Recall, F1-Score, and FPR. The Proposed LSTM model has shown to be better in terms of accuracy related measurements with the false positive rate being significantly lower than the current methods.

V. CONCLUSION

The noise-robust and attention-controlled DL system of network intrusion detection presented in

this research has been shown to handle the issues of noisy traffic, redundant features, and sophisticated attack patterns in a temporal manner. Through the proposed ANR-DP mechanism, the dynamic stabilization of network traffic characteristics was proposed through the repression of noise and maintenance of attack-relevant patterns, which resulted in enhanced data consistency. HRAFS strategy also promoted the reliability of detection by identifying security-critical features, which were identified by recursive elimination and attention-based relevance learning. Through the use of an LSTM-based classifier, the framework was able to identify both short term anomalies and long term intrusion patterns of a contemporary network traffic. Widespread testing of the CICIDS2017 dataset has shown that the given approach is even more successful in comparison with the current models of DL with regard to accuracy, robustness, stability of convergence, and reduction of false positives. On the whole, the findings substantiate the idea that adaptive preprocessing and attention-based feature selection can be used to enhance DL-based intrusion detection systems to be deployed in the real world. The work moving forward will be directed at expansion of the framework to deal with zero-day attacks and highly unbalanced datasets. Also, the real-time deployment as well as lightweight optimization to edge and IoT environments will be discussed.

REFERENCE

- [1] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021, doi: 10.1016/j.procs.2021.05.025.
- [2] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, p. 111, 2021, doi: 10.3390/fi13050111.
- [3] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, p. 898, 2022, doi: 10.3390/electronics11060898.
- [4] V. Hnamte and J. Hussain, "DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system," *Telematics and Informatics Reports*, vol. 10, p. 100053, 2023, doi: 10.1016/j.teler.2023.100053.
- [5] A. Amouri, V. T. Alaparthi, and S. D. Morgera, "A machine learning based intrusion detection system for mobile Internet of Things," *Sensors*, vol. 20, no. 2, p. 461, 2020, doi: 10.3390/s20020461.
- [6] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting Internet of Things network attacks," *Alexandria Engineering Journal*, vol. 61, no.

- 12, pp. 9395–9409, 2022, doi: 10.1016/j.aej.2022.02.063.
- [7] A. Awajan, “A novel deep learning-based intrusion detection system for IoT networks,” *Computers*, vol. 12, no. 2, p. 34, 2023, doi: 10.3390/computers12020034.
- [8] M. A. Khan, “HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system,” *Processes*, vol. 9, no. 5, p. 834, 2021, doi: 10.3390/pr9050834.
- [9] S. M. Kasongo, “A deep learning technique for intrusion detection system using a recurrent neural networks based framework,” *Computer Communications*, vol. 199, pp. 113–125, 2023, doi: 10.1016/j.comcom.2022.12.010.
- [10] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, “CNN-based network intrusion detection against denial-of-service attacks,” *Electronics*, vol. 9, no. 6, p. 916, 2020, doi: 10.3390/electronics9060916.
- [11] B. Susilo and R. F. Sari, “Intrusion detection in IoT networks using deep learning algorithm,” *Information*, vol. 11, no. 5, p. 279, 2020, doi: 10.3390/info11050279.
- [12] V. Dutta, M. Choraś, M. Pawlicki, and R. Kozik, “A deep learning ensemble for network anomaly and cyber-attack detection,” *Sensors*, vol. 20, no. 16, p. 4583, 2020, doi: 10.3390/s20164583.
- [13] M. A. Ferrag, L. Shu, H. Djallel, and K. K. R. Choo, “Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0,” *Electronics*, vol. 10, no. 11, p. 1257, 2021, doi: 10.3390/electronics10111257.
- [14] B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, “Network intrusion detection model based on CNN and GRU,” *Applied Sciences*, vol. 12, no. 9, p. 4184, 2022, doi: 10.3390/app12094184.
- [15] O. A. Alzubi, J. A. Alzubi, M. Alazab, A. Alrabea, A. Awajan, and I. Qiqieh, “Optimized machine learning-based intrusion detection system for fog and edge computing environment,” *Electronics*, vol. 11, no. 19, p. 3007, 2022, doi: 10.3390/electronics11193007.