SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Dynamic Feature Extraction in Streaming Data for Real-Time Analytics

M. A. Prasanna[1], Yudhveer Singh Moudgil[2], Wu Sijie[3], Jyoti Mahur[4], Shalini E[5], Sivasankari V[6], Jyotsna Suryavanshi[7], Vijayalakshmi Pasupathy[8]

[1] Assistant Professor, Department of Computer Science and Engineering, K. Ramakrishnan College of Technology, Tiruchirappalli, Tamil Nadu, India. Email: mapsan1984@gmail.com
[2] Assistant Professor, Department of Computer Science & Engineering, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India. Email: socse.yudhveer@dbuu.ac.in ORCID: 0009-0002-4805-609X
[3] Research Scholar, School of Accounting, Lincoln University College, Malaysia. Email: suie.phdscholar@lincoln.edu.my
[4] Assistant Professor, Department of School of Engineering & Technology, Noida International University, Greater Noida, Uttar Pradesh, India. Email: jyoti.mahur@niu.edu.in
[5] Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: shalini@maher.ac.in
[6] Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: sivasankariv@maher.ac.in
[7] Department of Engineering, Science and Humanities, Vishwakarma Institute of Technology, Pune, Maharashtra 411037, India. Email: jyotsna.suryavanshi@vit.edu
[8] Associate Professor, Department of Computer Science and Engineering, Panimalar Engineering College, India. Email: pvijayalakshmi@panimalar.ac.in

Abstract
The explosive increase in traffic of networkcreatedby means ofInternet of Things (IoT) devices, industrial processes, and web applications has complicated cybersecurity, rendering traditional approaches obsolete due to their high computational costs and delays, and their lack of adaptability to emerging traffic patterns. Conventional techniques suffer from poor discrimination, and they fail to achieve high detection accuracies. To address these limitations, the proposed research introduces a Seven-Spot Ladybird–mutated Deep Recurrent Neural Network (SSL-DRNN), in which a DRNN is employed to capture sequential dependencies, while the SSLO algorithm is utilized for optimal feature selection. The model is evaluated using the Network Intrusion dataset from Kaggle, which containsmalicious andnormal network traffic samples 28,30,743. All input traffic data are normalized through Z-score method to achieve unit varianceandzero mean, and relevant features are further extracted using Incremental Principal Component Analysis (IPCA).The experiments prove that the approach performs well in separating normal traffic from malicious traffic, which achieves very high values of F1 score of 98.5%, accuracy of 98.4%, and False Positives Rate (FPR) of 0.012. The implementation of the model is done through the use of Python code written using TensorFlow, Keras, Scikit-Learn, NumPy, and Pandas. The suggested SSL-DRNN method greatly improves the anomaly detection in network traffic, which helps in classification accuracy, decreasing false positives, and increasing cybersecurity performance and reliability.

Keywords: Intrusion Detection, Attack Classification, Anomaly Detection, Internet of Things (IoT), Incremental Principal Component Analysis (IPCA)

1. Introduction

Networks are continuously growing in parallel with digital transformation, thereby exposing themselves to many challenges that exist in cyberspace. The emergence of sophisticated attack techniques makes it challenging to detect malicious activities through traditional methods. These trends have led to a growing demand for anomaly detection systems [1]. Growth of the Fifth Generation (5G) networkandInternet of

Things (IoT) is enhancing communication speeds but simultaneously increasing cybersecurity threats as well. The rise in cybersecurity issues likeDenial of Service (DoS)attacks, ransomware, scanning activities, and points to an immediate need for anomaly detection [2].IoT and Industrial Internet of Things (IIoT) technologies have grown tremendously in recent years within smart and industrial contexts. Nevertheless, the growth has resulted in many challenges, especially in security. There is a need for a context-aware approach that can perform anomaly detection [3].Modern cloud ecosystems make use of Application Programming Interfaces (APIs) extensively for inter-service interaction, thus making more areas susceptible to cyber-attacks and making vital infrastructure vulnerable to emerging threats. Conventional security mechanisms find it difficult to deal with contextual misuse of APIs [4].Industry 4.0, with its integration of all aspects of an industrial plantsenhance efficiency, but made it vulnerable to cyber-attacks on its systems. The need for anomaly detection within such Industrial Control Systems (ICS) networks is highly necessary owing to recent cases [5]. Detection of intrusion becomes very essential in finding the presence of network attacks, especially when combined with the development of Internet technology. Anomaly detection is difficult due to the increased complexity and dynamism of the traffic on the network, as well as the increased size of the network [6].

Research Aim: This research seeks to develop a reliable model for detecting network traffic intrusions based on efficient feature extraction, selection, and sequence learning that can ensure accurate classification of attacks and anomalies. This research utilize Incremental Principal Component Analysis (IPCA) and Seven-Spot Ladybird–mutated Deep Recurrent Neural Network (SSL-DRNN) approaches to minimize false detections, increase generalization ability, and detect anomalies accurately.

Research Organization: Research is systematically structured to have five primary segments. The first segment includes the introduction of the increasing issues in the field of network security. In the second segment, the existing methods of intrusion detection, along with their deficiencies, are discussed. The methodology for network traffic attack classification is depicted in the third segment. The fourth segment indicates performance evaluation and discussion, and the fifth segment highlights conclusion and future directions.

2. Related Work

Anomalies in network traffic can be efficiently identified by the Bidirectional Temporal Attention Convolutional Network (Bi-TACN) approach, which results in an accuracy of 88.51% and 82.5% on the Network Security Laboratory Knowledge Discovery and Data Mining dataset (NSL-KDD),and University of New South Wales Network-Based 2015 dataset (UNSW-NB15), respectively, but with minimal algorithmic requirements.Anomaly detection of trafficwas developed via Encrypted Traffic Self-Supervised Learning (ET-SSL)[8], and obtains 96.8% accuracy, but it is dependent on possible data and high-speed adaptation. Research [9] aimed at classifying IoT devices, as well as detecting the presence of an attack on smart homes, specifically for Broadband Service Providers (BSPs), by using Federated Learning (FL) approach. It has an attainment accuracy rate of greater than 80%. But, its restriction includes moderate accuracy rate and heterogeneity issue of smart home systems. Research [10] was held to detect Dos anomaly in Wireless Sensor Networks (WSN) utilizing Deep Convolutional Neural Network (DCNN),andPrincipal Component Analysis (PCA). Though it resulted up to 97.83% accuracy and 97.92% F1-Score, its dependency on the dataset and poor generalization, are its drawbacks. Development of a Machine Learning (ML)based intrusion detection system [11]for Data Mining (DM) using a hybrid model based on K-means clustering combined with the Sequential Minimal Optimization(SMO),achieved an efficiency equal to 97.4%, a reduction in false alarm rate by 12%; the disadvantage lies in a low level of instant usage. Anomaly detection was developed using theCyber Physical Systems (CPS) method based on signature, threshold, and Ensemble Learning (EL) behavioral detection. The model reaches an accuracy rate of 93.19% and an F1 score of 93.24%.It is limited to heterogeneity in CPS and depends on the configuration of the system for each domain [12].Detection of Distributed DoS attacks in 5G cellular networks [13] employsML multi-class classification approaches to decrease computational complexity. It attains an accuracy rate of up to 97.26%; however, it uses only one next-generation Node B (gNB). Research [14] detect intrusions in IoT devices in an edge computing environment by employing a lightweight Temporal Convolutional Network (TCN)with hybrid feature selection; it obtained 93.79% accuracy; however, the results can be dataset-dependent. Development of network intrusion detection techniques for unknown attacks in Unmanned Aerial Vehicles (UAVs) using the ZeroUAV approach. It provides 87.3% and 94.6% zero-shot and few-shot accuracies on the Unmanned Aerial

Vehicle – Network Intrusion Detection Dataset (UAV-NIDD), respectively. Oneof the limitations was the lower performance when dealing with new attacks [15].An intrusion detection model [16] was developed using a Convolutional Neural Network- Bidirectional Long Short-Term Memory- Autoencoder (CNN-BiLSTM-AE) approach, and it resulted in 98.3% of F1-scoreand98.1% of accuracy. Shortcomings include high complexity and threshold sensitiveness.

3. Methodology

Network intrusion detection and anomaly classification are achieved using the proposed model which combines ML and optimization approaches. In the first step, records of network traffic are acquired from the network intrusion dataset, which consists of normal and malicious traffic flows. In addition, the data is normalized through the Z-score method, and the dimensionality reduction technique IPCA selects only the significant attributes from data. The attributes are further provided to DRNN for sequential learning and SSLO for selecting the optimal features as shown in Figure 1.

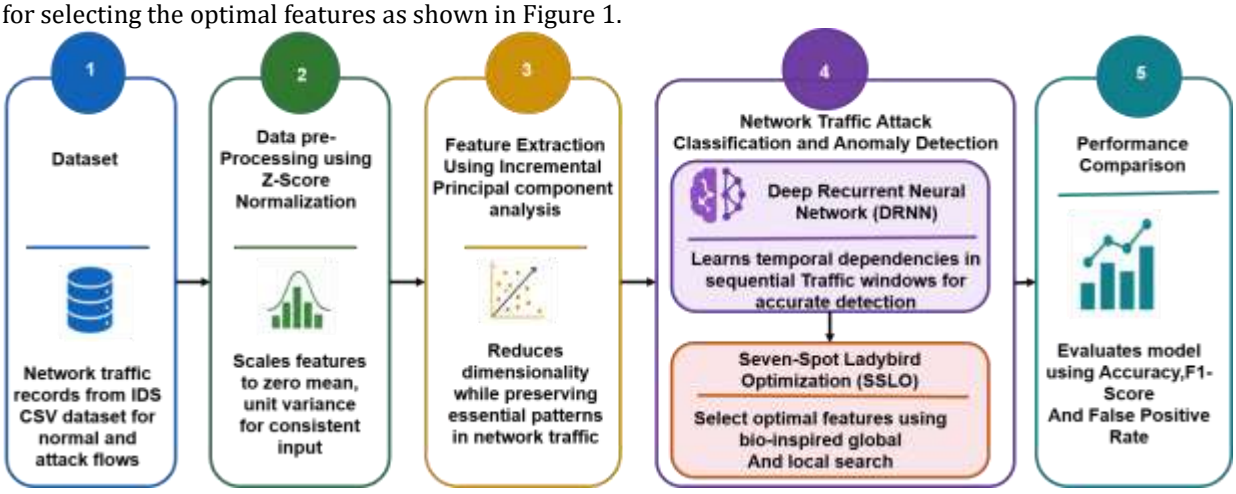


Figure 1: Methodology Flow of Proposed Architecture

3.1 Data Acquisition

The CSV file of the network intrusion dataset used for this research was collected from Kaggle (https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset). This dataset consists of approximately 28,30,743 samples of network flows with 79 features, depicting normal and abnormal characteristics of network traffic. The training and testing of the model is done using an 80:20 partitioning of this dataset.

3.2 Data Preprocessing using Z-score Normalization

Z-score normalization is a crucial preprocessing step for the network intrusion detection, as it standardizes numerical attributes by transforming them to a zero mean and unit variance. This prevents features with large value ranges, such as packet size and flow duration, from dominating the learning process. It reduces scale-induced bias, improves training stability, and supports balanced representation of normal and attack classes, which leads to improve the model's capability to acquire meaningful patterns in the traffic of network.

$$w' = \frac{w - \text{mean}(w)}{\text{Std}(w)} \quad (1)$$

In Equation (1), w indicates the feature value of the original input, w′ refers to the normalized output feature, mean and Std represent the average and standard deviation, and mean(w) and Std(w) indicate the feature value of the average and the standardization.

3.3 Feature Extraction using Incremental Principal Component Analysis (IPCA)

The IPCA technique helps to represent the data stream coming from the network intrusion dataset effectively. Through this method, the covariance pattern is continually updated to accommodate any additional sample data. IPCA is effective in reducing a very high-dimensional vector space into low dimensional vector. Thevariance and correlation data is conserved in the process as well. Through compression of redundant features, IPCA becomes an effective way of representing the dataset used for learning by ML algorithms to detect intrusions and anomalies.

$$||a_i||_2 = \sqrt{\sum_{j=1}^{n} |d_{ji}|^2} \quad (2)$$

In Equation (2), a denotes the feature vector,i indicates the vectorindex values, a_i refers to the feature vector of i,$||a_i||_2$ denotes the magnitude of the feature vector, $\sum$ represents the summation operator, n denote the count of features, j refers to the index of the features, $\sum_{j=1}^{n}$ depicts the sum of all elements,d indicates the value of the feature, d_{ji} represents the data values of the feature, and $|d_{ji}|^2$ depicts the magnitude of the squared feature,$\sqrt{}$ indicates the square root, and$\sqrt{\sum_{j=1}^{n} |d_{ji}|^2}$ represents the normalization of Euclidean function.

3.4 Network Traffic Attack Classification and IntrusionDetectionusingSeven-Spot Ladybird mutated Deep Recurrent Neural Network (SSL-DRNN)

The SSL-DRNN model applies biologically inspired feature extraction and sequential Deep Learning (DL) for identifying any possible attacks on the network intrusion dataset. The SSLO method carries out global and local searchfor the identification of sequential and important features by discarding redundancy. DRNN is used to make use of those sequentially optimized features by analyzing them for any anomalies present in the network traffic. The SSL-DRNN model is effective in discriminating normal traffic from malicious traffic through an efficient intrusion and anomaly detection system.

3.4.1 Deep Recurrent Neural Network (DRNN) for Network Traffic Attack and Intrusion Detection

DRNNs analyze network traffic by taking the data collected from the network intrusion dataset in the form of sequential vectors in fixed-size windows. This analysis of sequential vectors in windows takes into account both the temporal and contextual patterns of both the benign and attack traffic in the dataset. Class scores for various types of attacks and the traffic score are produced by the DRNN. These are later fused together to make accurate predictions on intrusions and anomalies in the network traffic.

$$z = \frac{1}{S} \sum_{s=1}^{S} z_s^K \quad (3)$$

In Equation (3), z refers to the aggregate score of the output, S indicates the total count of the sample, $\frac{1}{S}$ denotes the factor of average, $\sum$ refers to the summation function, s depicts the index value of the samples, $\sum_{s=1}^{S}$ indicates the sum of all the samples, andz_s^K refers to the score of class prediction.

3.4.2 Hyperparameter tuning with Seven-Shot Ladybird Optimization (SSLO)

The SSLO Algorithm is an optimization of meta-heuristic approach based on the foraging process of seven-spot ladybird beetles. It involves bothlocal search processand global search process, whereby the global search locates a subset of features, and the local search optimizes them into the desired solution. In this research, the SSLO algorithm has been used on the network intrusion dataset for choosing the most suitable subset of attributes. This ensures that there is no redundancy in the anomaly detection and attack classification of normal and abnormal traffic.

Search Space Partitioning: This SSLO technique finds D-dimensional hyperparameters of the network intrusion dataset by partitioning the dimensions into subspaces for effective intrusion detection**.**

$$N = \sum_{j=1}^{F} n_j \quad (4)$$

In Equation (4), N refers to the total count of the samples, $\sum$ indicates the summation operation,F denotes the total count of the features, j depicts the index of the features,$\sum_{j=1}^{F}$ refers to the sum over samples, n denotes the sample count, and n_j represents the data points for every attribute.

Initialization: Each ladybird is an SSLO solution candidate, and its hyperparameters are assessed using a fitness function based on the network intrusion dataset.

Selection of Best Ladybirds: During optimization of SSLO, there are three performance standards that determine the update process: individual best (obest), local best (kbest), and global best (hbest).

Mechanism for Position Update: In optimizing hyperparameters of DRNN, it makes use of both local and global search methods by SSLO, thereby enhancing network traffic detection precision.

$$X_j(q) = c_1 t_1 \left(obest_j - Y_j(q) \right) + \epsilon_1 \quad (5)$$

In Equation (5), X refers to the updated position, jdenotes the index of dimension, X_j indicates the value of dimension,cdepicts the coefficient of learning, $c_1$refers to the constant of adjustment, t indicatesthe factor of time, t_1 represents the duration of the step, orefers to the candidate, best indicates the best solution, $best_j$ denotes the value of elite feature, Y refers to the existing location, $Y_j$represents the active state of dimensional

index, q denotes the step of iteration, $Y_j(q)$ indicates the alternative of current solution, ϵ refers to the term of disturbance, and ϵ_1 indicates the constant of small disturbance.

$$Y_j(q+1) = Y_j(q) + X_j(q) \quad (6)$$

$$X_j(q) = c_2 t_2 \left(kbest - Z_j(q)\right) + \epsilon_2 \quad (7)$$

In Equation (6), qand $q+1$ represents the step of current iteration and future iteration, $Y_j(q)$and $Y_j(q+1)$indicates the alternative of ongoing and upcoming solution,X refers to the updated position, j denotes the index of dimension, X_j indicates the value of dimension, and $X_j(q)$ refers to position of new candidate.In Equation (7), c depicts the coefficient of learning,$c_2$indicates the coefficient of acceleration, t refers to the factor of time, t_2 represents the factor of random time,bestdenotes the solutionof best,krefers to thelocal position, Zindicates currentposition solution, Z_j refers to the position of solution in terms of dimensional index, $Z_j(q)$ depicts the position of the iteration step, ϵ refers to the term of disturbance, and ϵ_2 indicates the perturbation of random factor.

$$X_{max} = 0.2 \times (ub - lb) \quad (8)$$

In Equation (8), X refers to the updated position,max indicates the maximum value, X_{max} denotes the updated position of the maximum step, and ub and lb represent the upper and lower bound limits.

Exploration and Migration:SSLO is used to optimize the DRNN hyperparameters using both global and local searches with the help of migration, which improves feature selection and accuracy in network traffic anomaly detection

$$z_j'(z) = hbest + \varphi\omega \quad (9)$$

In Equation (9), z denotes the feature of the vector, z_j refers to the subset of the feature, z_j' indicates the subset of the updated feature, h refers to the state of the hidden solution, best represents the best solution, φ refers to the coefficient of the random factor, and ω denotes the factor of step. Algorithm 1 depicts the proposed SSL-DRNN model for detecting network intrusion and classifying anomalies based on feature extraction through IPCA and feature optimization through SSLO.

Algorithm: SSL-DRNN Intrusion Detection

Input: IDS CSV Dataset
Output: Accuracy, F1-Score, FPR
Begin
1. Load the IDS dataset
2. Apply Z-score normalization
3. Extract features using IPCA
4. Optimize features using SSLO
5. Train DRNN with optimized features
6. Classify traffic as normal or an attack
7. Evaluate performance using:
   - Accuracy
   - F1-Score
   - FPR
8. Compare with existing models
End

Hyperparameters of the proposed SSL-DRNN model: The SSL-DRNN system uses fine-tuned hyperparameters to provide a successful solution to network traffic anomaly detection. For the SSLO algorithm parameterization, hyperparameters like population size, max generation count, exploration and exploitation coefficient, mutation rate, seven-spot mutation rate, Gaussian noise rate, elitism rate, and fitness weights are considered to be useful in the feature selection phase. Hyperparameters like hidden layers, hidden units, sequence length, and activation functions are important for the neural network design of the DRNN algorithm. Learning rate, batch size, epoch count, optimizer, dropout, weight decay, and gradient clipping play important roles during training. Cross-entropy and L2 regularization functions can be used for loss calculation.

4. Result and discussion

The development and evaluation of the SSL-DRNN model using SSLO is implemented on a superior efficiency personal computer equipped with an Intel Core i9-13900K CPU featuring 24 cores and 32 threads, a base clock speed of 3.0 GHz, and a turbo boost capability up to 5.8 GHz. The system includes 64 GB DDR5 RAM, a 2 TB NVMe SSD, and NVIDIA GeForce RTX 4090 GPU with 24 GB GDDR6X memory, supported by high-speed internet connectivity. The programming setup comprises Python 3.11, Windows 11 Pro 64-bit, and Jupyter Notebook within Anaconda 2025 IDE. Key libraries include Matplotlib 3.10, Scikit-learn 1.5, NumPy 2.0, Plotly 6.0,Pandas 2.2, TensorFlow 2.16, SciPy 1.14, andKeras.

Explorative Data Analysis: Network traffic relationships, variations, and anomalies can be discovered using Figure 2, which illustrates correlation analysis and flow pattern visualization. Figure 2(a) depicts the values of the correlation coefficient for the characteristics of network traffic. The red color is used to represent high positive correlations, while blue colors signify negative correlations. Figure 2(b) shows the variation of network traffic sampling values. The positive and negative amplitudes represent the fluctuation in the network traffic behavior to show patterns, anomalies, and distribution characteristics.

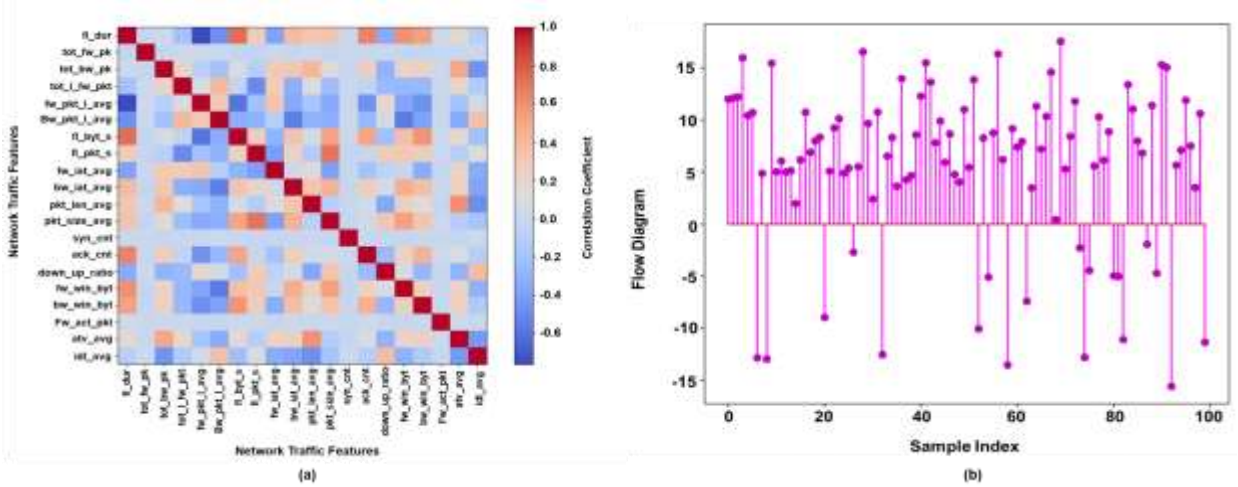


Figure 2: Visualization of (a) Correlation of Attributes in Network Traffic, and (b) Flow Behavior of Network Traffic Variation

The density of network flows, and contour mapping yields knowledge about traffic patterns and packet interactions is denoted in Figure 3. Figure 3(a) depicts the probability distribution of flow values. The presence of a noticeable peak represents the mode, and the additional variability reflects the nature of traffic and distribution, and Figure 3(b) shows the link between duration of flow and average length of packets. The colors indicate intensity variations, which reflect periodic patterns, interaction of features, and behavior of traffic in networks.

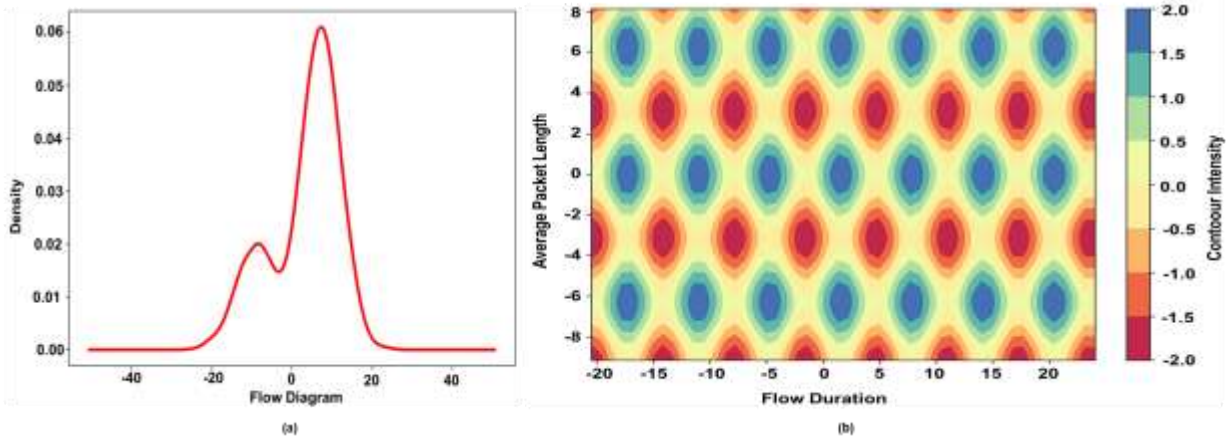


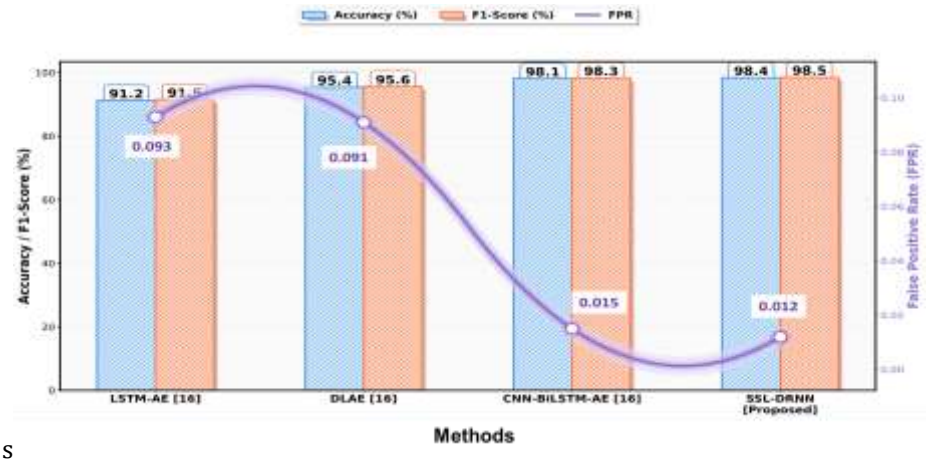
Figure 3: Visualization of (a) Network Traffic Flow Distribution Probability, and (b) Network Traffic Intensity Contour Mapping

Evaluation Metrics: The performance of the proposed SSL-DRNN architecture in this research is analyzed using 3 main evaluation parameters. **Accuracy** metric determines the percentage of correct classifications of instances belonging to network traffic. It evaluates the capacity of the model to classify normal and malicious behaviors. **F1 Score** evaluates the coordination amongrecalland precisionrate by considering bothfalse negatives and false positives. It helps to measure performance of classifier. **FPR** denotes the percentage of regular traffic categorized as attacks, showing the inclination to produce false alarms. All these measures help to understand performance of traffic intrusion detection of network.

Performance Analysis: The comparison of different intrusion detection models shown in Table 1 and Figure 4 reveals steady progress in terms of improving the performance of classifiers on benchmarks. While traditional methods includingLong Short-Term Memory Autoencoder (LSTM-AE) [16] and Deep Learning Autoencoder (DLAE)[16] offer relatively good outcome in terms of F1-Scoreand accuracy, hybrid models like CNN-BiLSTM-AE [16] deliver significantly better results with less frequent false positives. However, the proposed model of SSL-DRNN delivers outstanding performance metrics with 98.4% accuracy, 98.5% F1-Score, and 0.012 false positive rate. Such findings denote that the suggested approach has the ability to distinguishing attacks as well as malicious activity on networks with increased accuracy and reduced probability of errors.

Table 1: Evaluating Baseline Algorithms with the Proposed Model for Network Traffic Intrusion Detection

$Model$	$Accuracy$	$F1-Score$	FPR
$LSTM-AE$ [16]	91.2%	91.5%	0.093
$DLAE$ [16]	95.4%	95.6%	0.091
$CNN-BiLSTM-AE$ [16]	98.1%	98.3%	0.015
$SSL-DRNN$ $[Proposed]$	98.4%	98.5%	0.012



s

Figure 4: Comparative Analysis of Existing Models with the Proposed Model

All the baseline models, including CNN-BiLSTM-AE, DLAE, and LSTM-AE, were retrained using preprocessing and feature extraction approach on the network intrusion dataset shown in Table 2. The Z-score normalization technique was adopted for preprocessing, and the IPCA technique was used for extracting relevant features. The proposed SSL-DRNN model outperforms all baseline models in attack detection since it can effectively learn the temporal and contextual relationshipsamong network traffic features. Evaluation measures of the proposed model are accuracy (98.4%), F1 score (98.5%), and FPR (0.012).

Table 2: Retrained Evaluation of Baseline Models and Proposed Model Performance in Network Traffic Intrusion Prediction

Model	Accuracy	F1-Score	FPR

Model	Accuracy	F1-Score	FPR
LSTM-AE	88.5%	88.7%	0.120
DLAE	92.3%	92.5%	0.110
CNN-BiLSTM-AE	95.8%	96.0%	0.050
SSL-DRNN [Proposed]	98.4%	98.5%	0.012

Discussion

There are various shortcomings associated with current intrusion detection models, such as LSTM-AE [16], DLAE [16], and CNN-BiLSTM-AE [16], which make these models unsuitable for dealing with more complex and sophisticated traffic data. First, LSTM-AE [16] is not very effective at handling long-term dependencies among sequential data, which results in a poor detection rate and an increase in the number of false alarms. Though DLAE [16] addresses the issue of feature learning, the model continues to face problems related to poor generalization and high computational costs when dealing with highly dynamic and imbalanced traffic data. The CNN-BiLSTM-AE [16] has a higher degree of accuracy, but the deep hybrid nature of the network brings about high levels of complexity and computational burden along with extended training time. The suggested SSL-DRNN overcomes these challenges through optimal feature selection and recurrent learning, resulting in efficient pattern representation while eliminating redundancy. This ensures that there is improved adaptability and efficiency when implementing the model in terms of computational cost, thus ensuring enhanced detection accuracy for attacks and anomalies.

5. Conclusion

An SSL-DRNN model is introduced for effective detection of intrusions, attack and anomaly classification. The suggested model uses Z-score normalization along with IPCA and SSLO. With the help of optimized features and deep recurrent learning, it was able to efficiently diagnose the anomalies of the traffic in the network. The experimental results show promising performance in classifying network attacks, where the model reached a remarkable effectivenessusing a classification an F1 score of 98.5%,accuracy of 98.4%, and very low FPR of 0.012 on the network intrusion dataset. Compared to current methods, SSL-DRNN shows superiority in detecting attack types while minimizing false alarms. Thus, SSL-DRNN can be regarded as a reliable and efficient model for intrusion detection systems. However, the proposed SSL-DRNN approach was evaluated only on network intrusion dataset, which may affect its applicability in different network traffic environments. Potential future work may explore SSL-DRNN on different datasets to evaluate its generalization ability and adaptability to various attacks.

References

1. Yang, M. and Liu, C., 2026. XGA-E: an explainability-enhanced graph neural network for network traffic anomaly detection. Cybersecurity, 9(1), p.99.https://doi.org/10.1186/s42400-025-00487-x
2. Baldoni, S. and Battisti, F., 2025. Histogram-based network traffic representation for anomaly detection through PCA. Computer Networks, 265, p.111276.https://doi.org/10.1016/j.comnet.2025.111276
3. Stodt, F., Reich, C. and Theoleyre, F., 2026. Context-aware anomaly detection by community detection in the Internet of Things. Computer Communications, p.108414.https://doi.org/10.1016/j.comcom.2026.108414
4. Abibulaiev, A., Pukach, P. and Vovk, M., 2026. Context-Aware ML/NLP Pipeline for Real-Time Anomaly Detection and Risk Assessment in Cloud API Traffic. Machine Learning and Knowledge Extraction, 8(1), p.25.https://doi.org/10.3390/make8010025
5. Jiang, J.R. and Chen, Y.T., 2022. Industrial control system anomaly detection and classification based on network traffic. IEEE Access, 10, pp.41874-41888.https://doi.org/10.1109/ACCESS.2022.3167814
6. Ness, S., Eswarakrishnan, V., Sridharan, H., Shinde, V., Janapareddy, N.V.P. and Dhanawat, V., 2025. Anomaly detection in network traffic using advanced machine learning techniques. IEEE Access, 13, pp.16133-16149. https://doi.org/10.1109/ACCESS.2025.3526988
7. Wang, F., Huang, Y. and Shi, Y., 2026. Bidirectional Temporal Attention Convolutional Networks for High-Performance Network Traffic Anomaly Detection. Information, 17(1), p.61.https://doi.org/10.3390/info17010061

8. Sattar, S., Khan, S., Khan, M.I., Akhmediyarova, A., Mamyrbayev, O., Kassymova, D., Oralbekova, D. and Alimkulova, J., 2025. Anomaly detection in encrypted network traffic using self-supervised learning. Scientific Reports, 15(1), p.26585.https://doi.org/10.1038/s41598-025-08568-0
9. Rahman, M.M., Bouhafs, F., Hoseini, S.A. and Den Hartog, F., 2026. FedHome: A Federated Learning Framework for Smart Home Device Classification and Attack Detection by Broadband Service Providers. Computer Networks, p.112040.https://doi.org/10.1016/j.comnet.2026.112040
10. Yao, C., Yang, Y., Yin, K. and Yang, J., 2022. Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network. IEEE Access, 10, pp.103136-103149.https://doi.org/10.1109/ACCESS.2022.3210189
11. Gadal, S., Mokhtar, R., Abdelhaq, M., Alsaqour, R., Ali, E.S. and Saeed, R., 2022. Machine learning-based anomaly detection using K-mean array and sequential minimal optimization. Electronics, 11(14), p.2158.https://doi.org/10.3390/electronics11142158
12. Jeffrey, N., Tan, Q. and Villar, J.R., 2024. Using ensemble learning for anomaly detection in cyber–physical systems. Electronics, 13(7), p.1391.https://doi.org/10.3390/electronics13071391
13. Kim, Y.E., Kim, Y.S. and Kim, H., 2022. Effective feature selection methods to detect IoT DDoS attack in 5G core network. Sensors, 22(10), p.3819.https://doi.org/10.3390/s22103819
14. Gao, W., Wang, M., Pei, Y., Li, F. and Wang, C., 2026. A lightweight multi-classification intrusion detection model for edge IoT networks. Electronics, 15(5), p.938.https://doi.org/10.3390/electronics15050938
15. Alshamrani, A. and Alghamdi, A.M., 2026. Zero-shot attack detection in UAV networks using foundation models. Alexandria Engineering Journal, 136, pp.105-124.https://doi.org/10.1016/j.aej.2025.12.065
16. Park, H., Shin, D., Park, C., Jang, J. and Shin, D., 2025. Unsupervised machine learning methods for anomaly detection in network packets. Electronics, 14(14), p.2779.https://doi.org/10.3390/electronics14142779