EFRL-IDS: An Intelligent Early Fusion Deep Learning Framework for Cybernetic Intrusion Detection Systems

Ram Naresh Sharma[1], Jitendra Singh Kushwah[2], Pritaj Yadav[3]
[1st]Department of Computer Science and Engineering, Rabindranath Tagore University, Bhopal, India.
Email-nram73927@gmail.com
[2nd]Department of Information Technology, Institute of Technology and Management, Gwalior, India.
Email-jitendra.singhkushwah@itmgoi.in
[3rd]Department of Computer Science and Engineering, Rabindranath Tagore University, Bhopal, India;
Email-yadavpritaj@gmail.com

Corresponding Author: Ram Naresh Sharma*
Email-nram73927@gmail.com

Abstract:

Purpose: This study presents an intelligent Early Fusion Representation Learning Intrusion Detection System (EFRL-IDS) designed to enhance cyber protection through intelligent computing and cybernetic optimisation. The system aims to address challenges in recognising evolving intrusion patterns by integrating adaptive feature fusion, self-learning feedback, and hybrid deep learning techniques to intelligent decision-making in dynamic network environments.

Methodology: The suggested EFRL-IDS uses early data fusion to combine different features obtained from the UNSW-NB15 dataset, which ensures a complete picture of how the network behaves. The SMOTE-ENN method fixes class imbalance, and feature selection is done by Mutual Information, Chi-Squared, or Random Forest Importance. Random Forest is the most important for finding attacks. The hybrid deep architecture combines a pretrained Autoencoder to unsupervised representation learning, Convolutional Neural Networks to extracting spatial features, multi-head attention for improving context, and BiGRU layers for modelling temporal sequences.

Findings: The EFRL-IDS has a test accuracy of 86.91% and a validation accuracy of 87.01%, which is much better than traditional Early-Fusion Deep Learning models. Better precision, recall, or F1-scores show that it can find both. Types of common and rare assaults that show a lot of generalisation or adaptive learning across different traffic patterns.

Originality: EFRL-IDS introduces a cybernetically inspired, intelligent fusion-based deep learning framework that unifies data fusion, feature selection, and adaptive feedback learning. This model offers a scalable and interpretable solution for modern network security systems, contributing to the advancement of intelligent computing and autonomous cybernetic intrusion detection research.

Keywords : Intrusion Detection System (IDS), Early Data Fusion, Hybrid Deep Learning, Cybernetic Intelligence, Feature Selection, Network Security, Intelligent Computing

How to cite this article: Sharma RN, Kushwah JS, Yadav P. EFRL-IDS: an intelligent early fusion deep learning framework for cybernetic intrusion detection systems. Int J Drug Deliv Technol. 2026;16(73s): 821-833. DOI: 10.25258/ijddt.16.73s.90

1 Introduction

In the contemporary era of pervasive digital connectivity, societies increasingly depend on intelligent and interconnected networks to sustain critical infrastructures, business ecosystems, and daily communications. This transformation driven by e-commerce, financial systems, and smart governance has simultaneously intensified exposure to complex cyber threats. Among these, network intrusions represent a critical challenge that undermines data integrity, disrupts control systems, and compromises organiza- tional decision processes [1][3]. As cyberattacks becoming more complicated, we need defensive systems that can adapt, think for themselves, or regulate themselves that go beyond the limits of static or rule-based security systems.Traditional Intrusion Detection Systems (IDS) have used

deterministic or signature-based criteria to discover known attack patterns, but they have trouble adapting to new and changing invasions. [4][8]. Anomaly-based IDS approaches delineate normative network behaviour and detect irregularities; nevertheless, they often produce a significant volume of false positives owing to inadequate contextual understanding. [9][11]. To tackle these shortcomings, the integration of advanced computer paradigms, particularly machine learning (ML), deep learning (DL), or cybernetic feedback mechanisms, has become a vital emphasis in current cybersecurity research. [13][15]. These paradigms enable IDS frameworks to autonomously learn, adapt, and enhance their decision limits within dynamic network environments, hence establishing a foundation for intelligent cyber defence systems. Despite the progress enabled by ML and DL, standalone models often face chal- lenges related to high-dimensional feature spaces, imbalanced class distributions, and the need for rapid yet interpretable decision-making [15][22]. To mitigate these issues, researchers have explored hybrid and fusion-based intelligent architectures that com- bine the strengths of multiple computational paradigms within a unified cybernetic framework. Among these, data fusion has emerged as a vital strategy to aggregate heterogeneous traffic features and form holistic situational awareness. Particularly, early fusion, which integrates multi-dimensional features at the input stage, enables the system to capture correlated patterns across diverse network attributes thereby enhancing discriminative power and decision reliability [15][22]. Benchmark datasets such as UNSW-NB15, developed by the Australian Centre for Cyber Security, offer realistic testing grounds for such frameworks, containing both benign and malicious traffic across multiple attack categories (DoS, Exploits, Reconnaissance, Shellcode, etc.) [15][22]. However, the dataset?s high feature dimensionality and class imbalance necessitate preprocessing strategies such as SMOTE (Synthetic Minority Oversam- pling Technique) and ENN (Edited Nearest Neighbours) to ensure equitable class representation. In parallel, feature selection techniques like Mutual Information, Chi- Squared tests, and Random Forest ranking help eliminate redundant attributes and emphasize the most informative features, improving both accuracy and interpretability [21][26]. Against this backdrop, the present study proposes a Hybrid Intelligent Deep Learning Based Intrusion Detection System (HIDL-IDS) structured around cyber- netic feedback and intelligent control principles. The architecture integrates multiple modules: a pretrained autoencoder for dimensionality reduction and denoising, convo- lutional neural networks (CNNs) to extract spatial dependencies, multi-head attention mechanisms to emphasize contextual relations, and bidirectional gated recurrent units (Bi-GRU) to capture temporal dynamics [21][26]. These interconnected components collectively establish a self-optimizing cybernetic loop, allowing the system to adap- tively respond to evolving network states and attack behaviors. To further enhance resilience, the model incorporates Focal Loss during training, which prioritizes difficult- to-classify minority samples and reduces the dominance of major classes. A staged training strategy, beginning with autoencoder pretraining followed by multi-phase fine- tuning, supports robust generalization and stability. By uniting early fusion, feature selection, and deep hybrid architectures under an intelligent cybernetic design, this study contributes toward the next generation of adaptive, scalable, and interpretable IDS frameworks. The proposed system addresses the limitations of traditional and ML- based IDS and advances the broader vision of intelligent computing in cyber defense, ensuring security for heterogeneous, dynamic network environments [29][35].

2 Literature Review

The evolution of Intrusion Detection Systems (IDS) reflects a progressive transition from static rule-based mechanisms to intelligent, adaptive, and cybernetic architec- tures capable of learning and self-optimization. Over the past decade, researchers have explored diverse computational paradigms ranging from traditional machine learn- ing to deep learning and hybrid intelligent systems to strengthen network resilience against dynamic cyber threats.

2.1 Classical Machine Learning Based IDS

Early research laid the groundwork for data-driven detection by showing that even standard algorithms might find unusual behaviour in network traffic. Mishra and Mishra (2023) [36] utilised classical classification methods on the KDD 99 dataset to successfully tell the difference between good and bad activities. Their work was one for the first attempts to make decision support automatic in intrusion detection. Similarly, Mandru et al. (2022) [37] extended this direction by employing Deep Neural Networks (DNNs) with varying architectures and learning rates on KDDCup 99, revealing that deeper network layers outperform traditional ML models by recognizing novel and complex attack patterns. However, such models were still constrained by class imbalance, high-dimensionality, and limited contextual adaptability key challenges in the design of cybernetically intelligent systems.

2.2 Advanced Clustering and Optimization Techniques

To improve accuracy and cut down on false positives, Alfoudi et al. (2022) [38] came up with a hyper-clustering method that combines DBSCAN with cosine similarity or dynamic cluster evolution. This unsupervised paradigm showed a big improvement in how compact and adaptable clusters were across benchmark datasets. These optimization-driven methods are similar to the self-organising concepts that are at the heart of cybernetic systems, which change dynamically based on changes in input.

2.3 Hybrid and Intelligent Architectures for IoT and Network Environments

As cyber-physical and IoT infrastructures grew, IDS research shifted to hybrid and intelligent designs that could work in distributed contexts with limited resources. Banaamah and Ahmad (2022) [39] integrated machine learning (ML) and deep learning (DL) methodologies to fortify IoT networks against Denial of Service (DoS) attacks, utilising lightweight Python-based tools to facilitate practical deployment, a crucial advancement towards the integration of intelligent systems in real-world applications. Liu et al. (2021) [40] similarly introduced a Spark-based hybrid IDS that combines K-Means clustering and Random Forests for initial classification, and then uses CNN and LSTM layers for deep sequential analysis. Their method, tested on NSL-KDD and CIC-IDS2017, showed how distributed artificial intelligence may be used to find intrusions, which is a big focus of cybernetics research.

2.4 Deep Learning Driven Hybrid Cybernetic Systems

The field has changed to multi-layered hybrid models that combine feature extraction, dimensionality reduction, or adaptive classification. Ugendhar et al. (2022) [6] presented a multi-layer deep Intrusion Detection System (IDS) that incorporates preprocessing, autoencoders, and classifiers, attaining an accuracy of 96.7%, however it has not been validated in real-time scenarios. Karthiga et al. (2022) [7] created a hybrid Intrusion Detection System (IDS) that uses ANFIS (Adaptive Neuro-Fuzzy Inference System) to find existing threats and CNN to find new assaults. It was more than 98% accurate on CIC-IDS2017. Although confined to VANET applications, their methodology demonstrated the amalgamation of soft computing with deep learning, a distinctive attribute of cybernetic system design. Wang et al. (2022) [8] used autoencoders and Isolation Forests to find new threats to IoT security. Dat-Thinh et al. (2022) [9] built a collaborative Intrusion Detection System (IDS) that uses federated intelligence and got an accuracy rate of 99.68%. These models demonstrate that individuals are increasingly focussing on intelligent coordination and distributed learning. But there are still issues with latency, scalability, and interoperability, which are all needed for cybernetic systems to perform successfully.

2.5 Intelligent Computing, Optimization, and Emerging Directions

Numerous complementary studies have advanced this domain by providing extensive intelligent computing perspectives. Islam et al. (2021) [41] performed a comparative examination of diverse machine learning and deep learning models employing IoT datasets, illustrating that deep architectures frequently outperform shallow models, however issues with explainability and lightweight deployment remain. Kushwah (2022) examined proxy cache optimisation, demonstrating that LRU AVL caching techniques improved secondary memory access times compared to conventional methods [42]. Singh Kushwah (2022) broadened the applications for supervised machine learning, illustrating that Decision Trees function as efficient dual-purpose tools for both regression and classification [43]. Pathak (2022) introduced blockchain-based secure digital record systems, highlighting the incorporation of intelligent control and trust mechanisms for secure data management [45]. Ghuraiya (2024) conducted a comprehensive analysis of swarm intelligence algorithms, such as PSO, ACO, and BA, assessing their adaptive and collaborative decision-making functionalities inside cyber-physical systems [46]. Current IDS research has advanced from conventional machine learning to deep hybrid models; yet, significant shortcomings remain in adaptive intelligence and data fusion. Many systems do not have self-optimization based on input, and it can be hard to appropriately connect different network pieces. This work introduces a Hybrid Deep Learning-Based Intrusion Detection System that utilises early data fusion and advanced feature selection to enhance contextual awareness and interpretability. The model achieves intelligent adaptation, robust detection, and dynamic decision-making by integrating Autoencoders, CNN, Attention, or Bi-GRU into a cybernetic feedback framework. This effectively tackles issues of scalability, heterogeneity management, and real-time responsiveness in modern intrusion detection systems.

3 Research Methodology

The suggested study framework uses an

intelligent and cybernetic method for intrusion detection that combines adaptive hybrid deep learning. The data comes from the UNSW-NB15 dataset or goes through a lot of preparation, which includes getting rid of duplicates, normalising, and encoding categorical variables. The use of SMOTE-ENN helps to fix class imbalance so that all attack categories are fairly represented. A fusion-driven feature engineering pipeline uses Mutual Information, Chi-Squared, and Random Forest significance to find the most distinguishing features and make a coherent, high-quality feature space. Unsupervised representation learning is based on a pretrained autoencoder that works with regular network data. After being compressed, the embeddings are passed to a hybrid deep learning model that combines 1D CNN, multi-head attention, or Bi-GRU layers. This idea includes geographical, contextual, and sequential interconnection, which lets smart systems adapt to changing network patterns. The classifier uses Focal Loss to effectively deal with class imbalance. A dynamic feedback technique uses adaptive rate scheduling and checkpoint-based optimisation to keep learning going and make sure that it keeps getting better. This hybrid architecture, which is based on cybernetics, focusses on data fusion, adaptive intelligence, or self-optimization. The end result is an intrusion detection system that can think for itself and provide decision assistance in complicated, real-world network environments.

3.1 Data Preprocessing

The UNSW-NB15 dataset went through a strict cybernetic preprocessing approach to make sure that the data was reliable and the analysis was consistent. To avoid model bias and overfitting, redundant or duplicate instances were deleted at first. To make features more relevant, non-informative elements like unique IDs or binary class labels were removed. More attention was paid to the attack cat property, which makes it easier to sort attacks into different classes. This makes the learning process more subtle and intelligent, which is necessary for finding complex cyber threats. We label-encoded all categorical variables, such protocol, service, and state, into numbers while keeping their meanings separate. This made it possible for intelligent computing. The attack cat column was also encoded to make it easier to sort things into more than one class. After encoding, features were grouped by data type to make context-aware processing easier. For example, continuous (float-type) variables showed how traffic was changing, and integer-type metrics showed how communication was happening in a more discrete way. To stop data from leaking during training, the target variable was split up. This structured preprocessing pipeline makes a feature space that is ready for integration and works well with adaptive learning and cybernetic optimisation in the proposed hybrid deep learning framework.

3.2 Feature Engineering

The first step in feature engineering was to encode the categorical variables or the target column to the attack cat with labels. This turned all of the important features into numbers so that hybrid learning models could handle them more quickly. We standardised continuous (float-type) features for uniform scaling, while we kept discrete (integer-type) features to provide network event counts and categorical metrics. The various feature sets were then merged by early data fusion, creating a single, context-rich feature matrix. This fusion-based representation lets the model learn spatial, temporal, and contextual patterns all at once, which makes adaptive decision-making better and the cybernetic hybrid IDS work better overall.

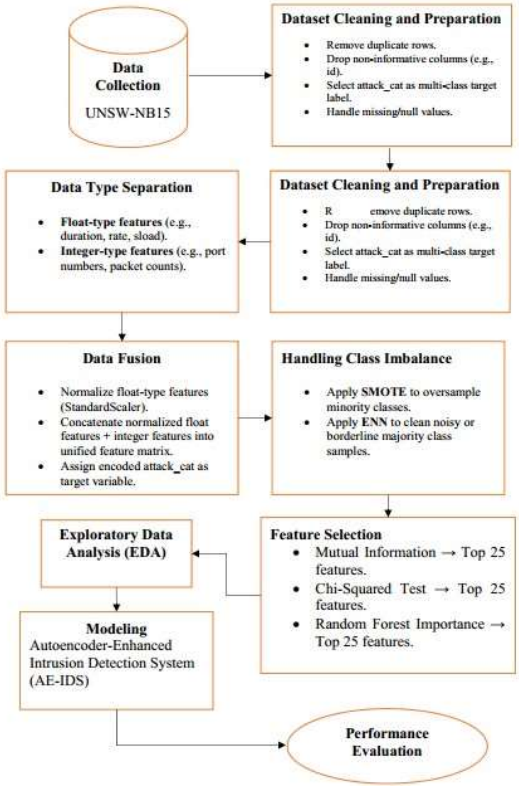

Fig.1 Flowchart of proposed approach.

Handling Class Imbalance with SMOTE-ENN

The UNSW-NB15 dataset showed a lot of class imbalance, with some assault types being very under-represented. This made it hard to train models

effectively. To make the distributions even, synthetic instances were created for minority classes using an SMOTE-based oversampling method. After that, Edited Nearest Neighbours (ENN) was used to get rid of samples from the majority classes that were noisy or unclear, which made the dataset better. This hybrid oversampling-cleaning approach improved the quality of the feature space and made it more representative of all types of attacks. By ensuring fair learning, the system may dynamically recognise different assault patterns, making intrusion detection that is aware of the environment and cybernetically informed easier.

Data Early Fusion of Numerical Features

In order to provide a full picture for network traffic, continuous (float-type) or discrete (integer-type) data were combined into one input space. Using z-score normalisation, continuous features were normalised to have a mean of 0 or a standard deviation of 1. This reduced the effect of large-scale attributes that were too strong. After normalisation, float or integer features were combined at the input level, creating a single feature matrix that was richer in context. The encoded labels to attack cats made up the target vector, which allowed supervised hybrid learning for all types of attacks. This first fusion method makes it easier for adaptive pattern recognition or cybernetic feedback systems to work inside the proposed deep learning framework.

3.3 Feature's Selection Techniques

As an intelligent intrusion detection systems to work better, they need to be able to choose features efficiently. This will reduce dimensionality, make the system easier to understand, and speed up calculations. We used Mutual Information (MI), the Chi-Squared Test, and Random Forest feature importance on the preprocessed UNSW-NB15 dataset to find the features that best separate the two groups for hybrid deep learning. Mutual Information looks at how each attribute relates to the target class and highlights the ones that give the most information on attack categories. This approach found the top 25 features, such as dur, rate, sload, dload, sinpkt, sjit, service, spkts, dpkts, or ct srv dst. These traits summarise important geographical, temporal, and contextual trends in network traffic, making sure that the model focusses on the most useful variables to multi-class categorisation. The feature selection pipeline works with the pretreatment step, which comes after SMOTE-ENN class balancing or label encoding, to provide a strong, high-quality, and ready-to-fuse feature set. This improved input makes adaptive learning easier within the hybrid framework, which lets for quick and context-sensitive detection in a wide range of assault scenarios.

$$(X_{\text{balanced}}, y_{\text{balanced}}) = \text{SMOTEENN}(X, y), \quad (1)$$

where,

$$\text{SMOTE: } x' = x + \delta \cdot (x_{\text{nn}} - x), \quad \delta \sim U(0,1)$$

$$\text{ENN: remove } x_i \text{ if majority of k} - \text{nearest neighbors} \neq y_i \quad (2)$$

SMOTE generates synthetic samples for minority attack classes, while ENN removes noisy or borderline samples from majority classes, resulting in a balanced dataset ready for machine learning.

$$y_{\text{encoded}} = \text{target\_encoder}(y_{\text{balanced}}), \quad \text{AttackName}_c = \text{target\_encoder}^{-1}(c), \quad (3)$$

All class labels are encoded into integers, ensuring compatibility with machine learning models. Encoded labels can be mapped back to attack names for interpretation and analysis.

$$I(X_j; y_{\text{encoded}}) = \sum_{x_j \in X_j} \sum_{y_i \in y_{\text{encoded}}} p(x_j, y_i) \log \frac{p(x_j, y_i)}{p(x_j)p(y_i)}, \quad (4)$$

$$\mathcal{F}_{\text{MI}} = \{X_j | I(X_j; y_{\text{encoded}}) \text{ is among top} k\}, \quad k = 25, \quad (5)$$

Mutual Information evaluates the dependency between each feature and the encoded target class. The top 25 features with highest MI scores are selected, forming a robust and informative feature set for downstream intrusion detection modeling.

$$X_{\text{balanced}} \in \mathbb{R}^{n \times m}, \quad y_{\text{encoded}} \in \{0,1, \ldots, C-1\}^n$$

$$n = \text{number of samples}, \quad m = \text{number of features}$$

$$C = \text{number of attack categories}$$

Chi-Squared Test is a statistical technique that measures the correlation between features and class labels, requiring all features to be non-negative. The 25 features with the strongest associations, such as dload, spkts, dbytes, ct_srv_src, and ct_dst_ltm, were retained. This method ensures that features contributing significantly to class discrimination are prioritized, improving predictive performance while removing irrelevant attributes.

The balanced dataset obtained via SMOTE-ENN is used for Chi-Squared feature selection.

Negative feature values are replaced with zero to satisfy the method's requirements, and the top 25 features with the highest χ^2 scores are selected as the most discriminative for intrusion detection.

$$X_j^{\text{non-neg}} = \max(X_j, 0), \quad \forall j = 1,2,\dots,m \tag{6}$$

$$\chi_j^2 = \sum_{i=1}^{n} \frac{(O_{ij}-E_{ij})^2}{E_{ij}}, \quad \mathcal{F}_\chi = \{X_j | \chi_j^2 \text{isamongtop} k\}, \quad k = 25 \tag{7}$$

Chi-Squared feature selection requires non-negative features; negative values are replaced with zero. Each feature's dependency on class labels is quantified using the χ^2 statistic. The top 25 features with highest scores are selected as the most discriminative attributes for downstream classification.

$X_{\text{balanced}} \in \mathbb{R}^{n\times m}$, $y_{\text{balanced}} \in \{0,1,\dots,C-1\}^n$

$n = \text{numberofsamples}$, $m = \text{numberoffeatures}$

$C = \text{numberofattackcategories}$

Random Forest (RF) uses a group of decision trees to figure out how important a feature is by lowering its impurity. Key factors, including sbytes, smean, service, sttl, ct dst src ltm, and trans depth, were pinpointed. Random Forest finds factors that have a big effect on categorisation assessments and gives a wide range of predictors. A Random Forest classifier is trained on a balanced dataset provided using SMOTE-ENN. The ensemble to decision trees gives us feature importance scores, which we use to pick the 25 most important characteristics for quick intrusion detection.

$$\text{RF}_{\text{importance}(X_j)} = \frac{1}{T}\sum_{t=1}^{T} \Delta I_t(X_j), \quad \mathcal{F}_{\text{RF}} = \{X_j | \text{top} k \text{importancescores}\}, \quad k = 25 \tag{8}$$

Random Forest computes feature importance via impurity reduction across all decision trees. The top 25 features with highest importance scores are selected as the most discriminative features for downstream classification.

$X_{\text{balanced}} \in \mathbb{R}^{n\times m}$, $y_{\text{balanced}} \in \{0,1,\dots,C-1\}^n$

$n = \text{numberofsamples}$, $m = \text{numberoffeatures}$

$C = \text{numberofattackcategories}$

$T = \text{numberoftreesintheRandomForest}$

Table 1: Top 25 Features Selected by Different Techniques

Mutual Information	Chi-Squared Test	Random Forest
dur	dload	sbytes
rate	sjit	smean
sload	service	service
dload	spkts	sttl
sinpkt	dpkts	ct_dst_src_ltm
dinpkt	sbytes	ct_srv_dst
sjit	dbytes	ct_dst_sport_ltm
proto	sttl	ct_src_dport_ltm
service	dttl	ct_srv_src
spkts	sloss	sload
dpkts	dloss	ct_dst_ltm
sbytes	swin	dmean
dbytes	stcpb	trans_depth
sloss	dtcpb	ct_flw_http_mthd
dloss	dwin	ct_src_ltm
smean	smean	rate
dmean	dmean	dbytes
ct_srv_src	response_body_len	dur
ct_state_ttl	ct_srv_src	ct_state_ttl
ct_dst_ltm	ct_dst_ltm	proto
ct_src_dport_ltm	ct_src_dport_ltm	dload
ct_dst_sport_ltm	ct_dst_sport_ltm	sinpkt
ct_dst_src_ltm	ct_dst_src_ltm	dpkts
ct_src_ltm	ct_src_ltm	sloss
ct_srv_dst	ct_srv_dst	tcprtt

3.4 EDA

Exploratory Data Analysis (EDA) was performed on the UNSW-NB15 dataset to understand its structure and distribution. Initially, the dataset showed a big class imbalance, with some attack types, like Worms and Shellcode, not having enough examples. Figures 1 and 2 show how the courses were split up before and after the SMOTE-ENN approach was put into place. This way of balancing classes made sure that minority attack classes were well-represented during training. This is necessary for developing a multi-class security detection model that works well. The EDA shows that the dataset's distribution has changed from being very skewed to being more even. This is a better place to train models. Figures 1 and 2demonstrate how the classes in the UNSW-NB15 dataset were spread out before and after SMOTE-ENN was employed. This shows how the dataset went from having a very uneven distribution to having an even split of assault types.

3.5 Auto encoder Pre training on Normal Traffic

The proposed technique begins with Autoencoder pretraining on standard network data to identify inherent behavioural patterns. The Autoencoder gets a compressed latent representation that makes it easy to spot anomalies by separating samples that are tagged as "Normal." This method of unsupervised learning allows the model uncover weird things that happen because people are doing badly.

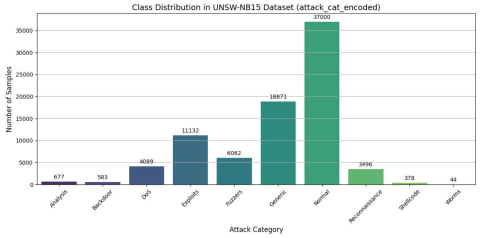

Figure 2: Class distribution in the UNSW-NB15 dataset before SMOTE-ENN

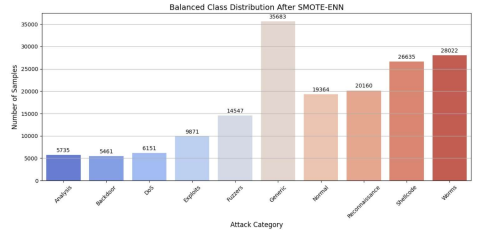

Figure 3: Class distribution in the UNSW-NB15 dataset after SMOTE-ENN

Autoencoders take inputs and compress them into a latent space, then put them back together. This helps kids understand how genuine networks work together. Early data fusion puts together different types of network properties before they are used. This makes the representation more full and useful. The Autoencoder learns to reduce reconstruction error by dividing standard samples into sets for training or testing. This makes it easier to find useful features. The pretrained encoder is preserved and incorporated to the hybrid deep learning system. This makes it easier to see trends in different kinds of attacks and makes classification more accurate.

Let the input feature vector representing normal traffic be:

$$\mathbf{x} \in \mathbb{R}^{25} \tag{9}$$

The encoder maps $\mathbf{x}$ to a latent representation $\mathbf{z} \in \mathbb{R}^{16}$:

$$\mathbf{h}_1 = \mathrm{ReLU}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \quad \mathbf{W}_1 \in \mathbb{R}^{32\times25}, \mathbf{b}_1 \in \mathbb{R}^{32} \tag{10}$$

$$\mathbf{z} = \mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2, \quad \mathbf{W}_2 \in \mathbb{R}^{16\times32}, \mathbf{b}_2 \in \mathbb{R}^{16} \tag{11}$$

The decoder reconstructs the input from $\mathbf{z}$:

$$\mathbf{h}_2 = \mathrm{ReLU}(\mathbf{W}_3\mathbf{z} + \mathbf{b}_3), \quad \mathbf{W}_3 \in \mathbb{R}^{32\times16}, \mathbf{b}_3 \in \mathbb{R}^{32} \tag{12}$$

$$\hat{\mathbf{x}} = \mathbf{W}_4\mathbf{h}_2 + \mathbf{b}_4, \quad \mathbf{W}_4 \in \mathbb{R}^{25\times32}, \mathbf{b}_4 \in \mathbb{R}^{25} \tag{13}$$

The reconstruction loss is measured using Mean Squared Error (MSE):

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N} \| \mathbf{x}_i - \hat{\mathbf{x}}_i \|_2^2 \tag{14}$$

Using gradient-based optimization (Adam), the network parameters $\theta = \{\mathbf{W}_1, \mathbf{b}_1, \ldots, \mathbf{W}_4, \mathbf{b}_4\}$ are updated as:

$$\theta \leftarrow \theta - \eta \cdot \hat{\nabla}_\theta \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) \tag{15}$$

After training, only the encoder is retained:

$$\mathbf{z} = f_{\text{encoder}}(\mathbf{x}) \tag{16}$$

This latent vector $\mathbf{z}$ is then used as a pretrained feature representation in a hybrid intrusion detection model. Early data fusion can combine $\mathbf{z}$ with other

network metrics to improve anomaly detection performance.

3.6 Hybrid Deep Learning Architecture for Cybernetic Intrusion Detection

The suggested hybrid deep learning model uses multiple types of neural networks to detect complicated patterns in network traffic that have to do with time, place, and context. It combines the pretrained Autoencoder encoder, convolutional layers, multi-head attention, or bidirectional GRU (Bi-GRU) to leverage the fact that the fused features work well together. The Autoencoder, which has been trained on standard data, makes latent embeddings z that depict what normal activity looks like. This is a good starting point for finding anomalies. Preliminary data fusion combines these embeddings with other network measures to create a single, contextually rich input space that makes the model more responsive to anomalies that could mean an assault. Dropout layers stop overfitting, while convolutional layers find localised spatial patterns that show short-term correlations. The multi-head attention approach makes the most important parts stand out, making it easier to focus on important temporal or spatial interactions. The Bi-GRU captures sequential dependencies in both directions, creating feature representations that take context into account for subsequent categorisation. The classifier has fully connected layers that use nonlinear activation or dropout to make predictions about different types of attacks. Focal Loss helps with class imbalance by focussing on samples from the minority class that are hard to classify. This makes it easier to find threats that are not well represented, like Worms or Shellcode, while still keeping the overall performance robust. The dataset has been preprocessed and has features chosen, and now it is split into three parts: training, validation, and testing. The training and validation subsets make it easier to learn in steps, while the test set checks how well the model works in the end. The model is trained for 100 epochs while checking its accuracy, precision, recall, or macro F1-score. A ReduceLROnPlateau scheduler changes the learning rates automatically when the validation performance levels off, and checkpointing keeps the model that has the highest validation F1-score. Consistent documentation of comprehensive measurements ensures a dependable evaluation of the hybrid architecture's adaptive, context-aware, and cybernetically informed intrusion detection capabilities.

Let the input feature vector be:

$$\mathbf{X} \in \mathbb{R}^d, \quad d = 25 \tag{17}$$

The pretrained Autoencoder encoder $f_{\mathrm{AE}}(\cdot)$ is used to extract latent features:

$$\mathbf{Z} = f_{\mathrm{AE}}(\mathbf{X}) \in \mathbb{R}^l, \quad l = 16, \tag{18}$$

The convolutional layer maps latent features to local feature maps:

$$\mathbf{C} = \mathrm{ReLU}(\mathrm{Conv1D}(\mathbf{Z})) \in \mathbb{R}^{F \times L}, \quad F = \mathrm{numfilters}, L = \mathrm{sequencelength} \tag{19}$$

The multi-head attention mechanism computes weighted feature interactions:

$$\mathbf{A} = \mathrm{MultiHeadAttention}(\mathbf{C}, \mathbf{C}, \mathbf{C}) \in \mathbb{R}^{F \times L} \tag{20}$$

The bidirectional GRU captures sequential dependencies:

$$\mathbf{H} = \mathrm{BiGRU}(\mathbf{A}) \in \mathbb{R}^{2h}, \quad h = 64 \tag{21}$$

Finally, the classifier maps the representation to class probabilities:

$$\widehat{\mathbf{Y}} = \mathrm{Softmax}(\mathbf{W}_c\mathbf{H} + \mathbf{b}_c), \quad \widehat{\mathbf{Y}} \in \mathbb{R}^{n_{\mathrm{classes}}} \tag{22}$$

The model is trained using Focal Loss:

$$\mathcal{L}_{\mathrm{focal}}(\widehat{\mathbf{Y}}, \mathbf{Y}) = -\frac{1}{N}\sum_{i=1}^{N} (1 - \hat{y}_i)^{\gamma} y_i \log(\hat{y}_i), \quad \gamma = 2 \tag{23}$$

During training, the network parameters are updated via gradient descent:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}_{\mathrm{focal}}(\widehat{\mathbf{Y}}, \mathbf{Y}) \tag{24}$$

Here, θ includes the pretrained encoder parameters, CNN, attention, BiGRU, and classifier weights. Early data fusion is achieved by combining $\mathbf{Z}$ with additional input features before CNN processing, enhancing the model's capability to detect anomalies in network traffic.

Table 2: Hyperparameters of the Proposed Hybrid Model

Component	Hyperparameter	Value / Setting
Autoencoder Encoder	Input Dimension	25
Autoencoder	Latent	16

Encoder	Dimension	
Autoencoder Encoder	Hidden Layer Dimension	32
Autoencoder Decoder	Hidden Layer Dimension	32
CNN Layers	Conv1 Filters	16
CNN Layers	Conv2 Filters	32
CNN Layers	Kernel Size	5
CNN Layers	Dropout Rate	0.2
Attention Layer	Type	Multi-head Attention
Attention Layer	Number of Heads	4
BiGRU Layer	Hidden Units	64
BiGRU Layer	Direction	Bidirectional
BiGRU Layer	Dropout	0.3
Classifier	Fully Connected Layer 1 Units	64
Classifier	Dropout	0.3
Classifier	Fully Connected Layer 2 Units	Number of Classes
Training	Optimizer	AdamW
Training	Learning Rate	0.001
Training	Loss Function	Focal Loss ($\gamma = 2$)
Training	Batch Size	64
Training	Epochs	100
Training	LR Scheduler	ReduceLROnPlateau (patience=4, factor=0.5)

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (25)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (26)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (27)$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision+Rec}} \quad (28)$$

where,

$$TP = \text{TruePositives}$$
$$TN = \text{TrueNegatives}$$
$$FP = \text{FalsePositives}$$
$$FN = \text{FalseNegatives} \quad (29)$$

4 Results

The evaluation of the suggested hybrid Autoencoder + CNN + BiLSTM model utilising early data fusion demonstrates that feature selection is essential for improving intrusion detection effectiveness. Among the applied procedures, Random Forest-based feature selection gave the most consistent and superior outcomes, followed by Mutual Information, while Chi-Squared demonstrated somewhat inferior performance.The hybrid model, which used Mutual Information, showed strong generalisation across all datasets. It had training, validation, and validation accuracies of 0.8402, 0.8605, and 0.8606, respectively, and a balanced test F1-score of 0.7631. These results show that Mutual Information is quite good at finding the most useful parts, which makes it easier to The model to accurately differentiate between normal or attack traffic patterns. Adding early data fusion to Autoencoder-derived latent representations speeds up this process, which improves anomaly detection in a variety of network situations. The Chi-Squared approach did not do very well, with testing and training accuracies of 0.6776 or 0.6857, respectively. Even though the precision was only 0.7791, the recall was only 0.5819 and the F1-score was only 0.6050, which shows that it wasn't very good at finding complex, non-linear patterns related to violence against minorities. This finding underscores the inadequacy of relying simply on statistical correlation methods for simulating high-dimensional network data. On the other hand, Random Forest feature selection gave the best results, with training, validation, and testing accuracies of 0.8537, 0.8701, and 0.8691, with a test F1-score of 0.7737. The balance between precision and recall shows that it works well for finding both common and rare types of assaults. The hybrid Autoencoder + CNN + BiLSTM architecture can use both linear and non-linear relationships in the data because it combines feature importance from Random Forest with early data fusion. This makes the architecture more accurate, more generalisable, and more resistant to class imbalance. This research shows how useful hybrid learning and information fusion methods can be for getting reliable intrusion detection in networks that change all the time.

Table 3: Performance Comparison of Feature Selection Methods using Early Data Fusion on the Proposed Hybrid Model

Feature Method	Dataset	Accuracy	Precision	Recall	F1-Score
Mutual Information	Train	0.8402	0.7599	0.7369	0.7412
	Val	0.8605	0.7906	0.7652	0.7688
	Test	0.8606	0.7887	0.7584	0.7631
Chi-Squared Test	Train	0.6776	0.7442	0.5801	0.6019
	Val	0.6773	0.7792	0.5802	0.6043
	Test	0.6857	0.7791	0.5819	0.6050
Random Forest Feature	Train	0.8537	0.7751	0.7536	0.7583
	Validation	0.8701	0.8070	0.7793	0.7849
	Test	0.8691	0.7969	0.7677	0.7737

Table 4: Comparative Analysis Between Existing and Proposed Models

Models	Test Accuracy	Validation Accuracy	References
Early-fusion DL model	76.47	84.44	[40]
Random Forest Importance AE-IDS	86.91	87.01	–

This study use Random Forest-based feature importance to assess the performance of a baseline Early-Fusion Deep Learning model in comparison to the suggested hybrid deep learning technique. The comparison looks at test or validation accuracy to see how well the model can generalise. The baseline Early-Fusion DL model got a test accuracy of 76.47% and a validation accuracy of 84.44%, which shows that it can generalise rather well. Early fusion makes it easier to merge information before classification, but the model had problems figuring out complicated and different intrusion patterns, especially when it had to deal with traffic situations it wasn't used to.

In contrast, the proposed hybrid model, which integrates Random Forest-based feature selection with a pretrained Autoencoder for dimensionality reduction, demon- strated significantly improved performance. It achieved 86.91% test accuracy and 87.01% validation accuracy, highlighting enhanced learning stability and robustness. The observed improvement results from the hybrid architecture?s ability to jointly learn spatial, contextual, and temporal dependencies while filtering out redundant fea- tures. This comparative analysis confirms that the proposed hybrid approach offers superior effectiveness and generalization for intrusion detection across diverse network environments.

5 Conclusion

A combination of the UNSW-NB15 dataset and a cybernetically inspired data-driven approach, this paper presents EFRL-IDS, an advanced hybrid deep learning system for smart intrusion detection. To deal with differences and imbalances in network traffic, the proposed system focusses on thorough data pretreatment, initial data fusion, and adaptive class balancing using the SMOTE-ENN technique. We looked at three ways to choose features: Mutual Information, Chi-Squared, and Random Forest Importance. Random Forest was the best because it helped reduce the number of dimensions while keeping important information for multiclass intrusion detection. The EFRL-IDS model has a pretrained Autoencoder or unsupervised representation learning. Then, CNN, multi-head attention, and BiGRU modules work together to capture spatial, contextual, and temporal connections. This architecture makes learning faster and sets up a cybernetic feedback loop that helps people make decisions that adapt to changing network conditions. The model had a validation accuracy of 87.01% and a test accuracy of 86.91%. It was more accurate and resilient than typical early-fusion structures. The evaluation findings show that the system works the same for all types of attacks. The higher F1-scores mean that it is better at finding attacks on minority groups. When you combine early data fusion, hybrid representation learning, and smart

optimisation, you get something that is both scalable and helpful in real life. EFRL-IDS improves intelligent computing and cybernetic systems for safe digital spaces. It lays the groundwork for self-adaptive, interpretable, and effective intrusion detection systems that follow the principles of cybernetic intelligence and have a lot of potential for use in dynamic cybersecurity infrastructures.

Author Contribution Statement

RNS conceived and designed the research. RNS performed the code implementation and drafted the manuscript. JSK and PY provided guidance, supervision, and critical feedback, and assisted in improving the manuscript. All authors read and approved the manuscript.

Funding Declaration

The authors declare that no funding was received from any organization, institution, or agency for the design, execution, analysis, or publication of this research work. This study was conducted independently without any financial support or external sponsorship.

6 Bibliography

1. N. Yadav, S. Pande, A. Khamparia, and D. Gupta, Intrusion Detection System on IoT with 5G Network Using Deep Learning, Wirel. Commun. Mob. Comput., vol. 2022, 2022, doi: 10.1155/2022/9304689.

2. S. Tharewal, M. W. Ashfaque, S. S. Banu, P. Uma, S. M. Hassen, and M. Shabaz, Intrusion Detection System for Industrial Internet of Things Based on Deep Reinforcement Learning, Wirel. Commun. Mob. Comput., vol. 2022, 2022, doi: 10.1155/2022/9023719.

3. S. K. B. Sangeetha, P. Mani, V. Maheshwari, P. Jayagopal, M. Sandeep Kumar, and S. M. Allayear, Design and Analysis of Multilayered Neural Network-Based Intrusion Detection System in the Internet of Things Network, Comput. Intell. Neurosci., vol. 2022, 2022, doi: 10.1155/2022/9423395.

4. A. K. Pandey, D. Jain, T. K. Gautam, J. S. Kushwah, S. Shrivastava, R. Sharma, and P. Vats, ’Tomato Leaf Disease Detection using Generative Adversarial Network-based ResNet50V2,’ Engineering Letters, vol. 32, no. 5, pp. 965-973, 2024.

5. D. Jain, A. K. Pandey, A. S. Chauhan, J. S. Kushwah, N. Saxena, R. Sharma, and V. D. P. Sambrow, ’ASA-LSTM-based brain tumor segmentation and classification in MRI images,’ Int. J. Adv. Technol. Eng. Exploration, vol. 11, no. 115, pp. 838-851, 2024, doi: 10.19101/IJATEE.2023.10102143.

6. A. Ugendhar et al.,A Novel Intelligent-Based Intrusion Detection System Approach Using Deep Multilayer Classification, Math. Probl. Eng., vol. 2022, 2022, doi: 10.1155/2022/8030510.

7. B. Karthiga, D. Durairaj, N. Nawaz, T. K. Venkatasamy, G. Ramasamy, and A. Hariharasudan, Intelligent Intrusion Detection System for VANET Using Machine Learning and Deep Learning Approaches, Wirel. Commun. Mob. Comput., vol. 2022, 2022, doi: 10.1155/2022/5069104.

8. Y. Wang, G. Sun, X. Cao, and J. Yang, An Intrusion Detection System for the Internet of Things Based on the Ensemble of Unsupervised Techniques, Wirel. Commun. Mob. Comput., vol. 2022, 2022, doi: 10.1155/2022/8614903.

9. N. Dat-Thinh, H. Xuan-Ninh, and L. Kim-Hung, MidSiot: A Multistage Intrusion Detection System for Internet of Things, Wirel. Commun. Mob. Comput., vol. 2022, no. December 2017, 2022, doi: 10.1155/2022/9173291.

10. E. M. Maseno, Z. Wang, and H. Xing, A Systematic Review on Hybrid Intrusion Detection System, Secur. Commun. Networks, vol. 2022, 2022, doi: 10.1155/2022/9663052.

11. A. Dahou et al., Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm, Comput. Intell. Neurosci., vol. 2022, pp. 1-15, 2022, doi: 10.1155/2022/6473507.

12. S. D. Alotaibi et al., Deep Neural Network-Based Intrusion Detection System through PCA, Math. Probl. Eng., vol. 2022, 2022, doi: 10.1155/2022/6488571.

13. X. Pu, Y. Zhang, and Q. Ruan, Optimization of Intrusion Detection System Based on Improved Convolutional Neural Network Algorithm, Math. Probl. Eng., vol. 2022, 2022, doi: 10.1155/2022/6762175.

14. J. Luo, H. Wang, Y. Li, and Y. Lin, Intrusion Detection System Based on Genetic Attribute Reduction Algorithm Based on Rough Set and Neural Network, Wirel. Commun. Mob. Comput., vol. 2022, 2022, doi: 10.1155/2022/5031236.

15. R. Chen, Design and Protection Strategy of Distributed Intrusion Detection System in Big Data Environment, Comput. Intell. Neurosci., vol. 2022, 2022, doi: 10.1155/2022/4720169.

16. X. Yang, G. Peng, D. Zhang, and Y. Lv, An Enhanced Intrusion Detection System for IoT Networks Based on Deep Learning and Knowledge

Graph, Secur. Commun. Networks, vol. 2022, no. Ml, 2022, doi: 10.1155/2022/4748528.

17. X. Chen and J. Pang, Temporal Logic-Based Artificial Immune System for Intrusion Detection, Wirel. Commun. Mob. Comput., vol. 2022, no. 1, pp. 1-9, 2022, doi: 10.1155/2022/4685754.

18. H. Wang, Q. Wei, and Y. Xie, A Novel Method for Network Intrusion Detection, Sci. Program., vol. 2022, 2022, doi: 10.1155/2022/1357182.

19. Z. Wang, J. Liu, and L. Sun, EFS-DNN: An Ensemble Feature Selection-Based Deep Learning Approach to Network Intrusion Detection System, Secur. Commun. Networks, vol. 2022, 2022, doi: 10.1155/2022/2693948.

20. W. Kim, J. Lee, Y. Lee, Y. Kim, J. Chung, and S. Woo, Vehicular Multilevel Data Arrangement-Based Intrusion Detection System for In-Vehicle CAN, Secur. Commun. Networks, vol. 2022, 2022, doi: 10.1155/2022/4322148.

21. H. Ma, J. Cao, B. Mi, D. Huang, Y. Liu, and S. Li, A GRU-Based Lightweight System for CAN Intrusion Detection in Real Time, Secur. Commun. Networks, vol. 2022, 2022, doi: 10.1155/2022/5827056.

22. Y. Niu, C. Chen, X. Zhang, X. Zhou, and H. Liu, Application of a New Feature Generation Algorithm in Intrusion Detection System, Wirel. Commun. Mob. Comput., vol. 2022, 2022, doi: 10.1155/2022/3794579.

23. J. Zhang and Y. Xiang, Research on Traffic Intrusion Detection Method Based on Deep Learning, Proc. - 2022 11th Int. Conf. Inf. Commun. Technol. ICTech 2022, vol. 2022, pp. 204-208, 2022, doi: 10.1109/ICTech55460.2022.00048.

24. L. Shi and K. Li, Privacy Protection and Intrusion Detection System of Wireless Sensor Network Based on Artificial Neural Network, Comput. Intell. Neurosci., vol. 2022, 2022, doi: 10.1155/2022/1795454.

25. F. Ye and W. Zhao, A Semi-Self-Supervised Intrusion Detection System for Multilevel Industrial Cyber Protection, Comput. Intell. Neurosci., vol. 2022, 2022, doi: 10.1155/2022/4043309.

26. G. M. H. Bashar, M. A. Kashem, and L. C. Paul, Intrusion Detection for Cyber-Physical Security System Using Long Short-Term Memory Model, Sci. Program., vol. 2022, 2022, doi: 10.1155/2022/6172362.

27. T. Rincy N and R. Gupta, Design and Development of an Efficient Network Intrusion Detection System Using Machine Learning Techniques, Wirel. Commun. Mob. Comput., vol. 2021, no. 1, 2021, doi: 10.1155/2021/9974270.

28. E. M. Zeleke, H. M. Melaku, and F. G. Mengistu, Efficient Intrusion Detection System for SDN Orchestrated Internet of Things, J. Comput. Networks Commun., vol. 2021, 2021, doi: 10.1155/2021/5593214.

29. N. Moukafih, G. Orhanou, and S. El Hajji, Neural Network-Based Voting System with High Capacity and Low Computation for Intrusion Detection in SIEM/IDS Systems, Secur. Commun. Networks, vol. 2020, 2020, doi: 10.1155/2020/3512737.

30. A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, Intrusion Detection System for Internet of Things Based on Temporal Convolution Neural Network and Efficient Feature Engineering, Wirel. Commun. Mob. Comput., vol. 2020, no. April, 2020, doi: 10.1155/2020/6689134.

31. M. Aljanabi, M. A. Ismail, and V. Mezhuyev, Improved TLBO-JAYA Algorithm for Subset Feature Selection and Parameter Optimisation in Intrusion Detection System, Complexity, vol. 2020, 2020, doi: 10.1155/2020/5287684.

32. L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, A Mean Convolutional Layer for Intrusion Detection System, Secur. Commun. Networks, vol. 2020, no. Ml, 2020, doi: 10.1155/2020/8891185.

33. P. Sun et al., DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system, Secur. Commun. Networks, vol. 2020, 2020, doi: 10.1155/2020/8890306.

34. J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, and H. Jingjing, Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms, Secur. Commun. Networks, vol. 2019, 2019, doi: 10.1155/2019/7130868.

35. V. V. Thang, F. F. Pashchenko, and A. H. Lashkari, Multistage system-based machine learning techniques for intrusion detection in WiFi network, J. Comput. Networks Commun., vol. 2019, p. 6576725, 2019, doi: 10.1155/2019/4708201.

36. N. Mishra and S. Mishra, A Novel Intrusion Detection Techniques of the Computer Networks Using Machine Learning, Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 5s, pp. 247-260, 2023.

37. D. B. Mandru, M. Aruna Safali, N. Raghavendra Sai, and G. Sai Chaitanya Kumar, Assessing Deep Neural Network and Shallow for Network Intrusion Detection Systems in Cyber Security, Lect. Notes Data Eng. Commun. Technol., vol. 75, no. May, pp. 703-713, 2022, doi: 10.1007/978-981-16-3728-5_52.

38. A. S. Alfoudi et al., Hyper clustering model for dynamic network intrusion detection, IET Commun., no. October, pp. 1-13, 2022, doi: 10.1049/cmu2.12523.

39. A. M. Banaamah and I. Ahmad, Intrusion Detection in IoT Using Deep Learning, Sensors, vol. 22, no. 21, 2022, doi: 10.3390/s22218417.

40. C. Liu, Z. Gu, and J. Wang, A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning, IEEE Access, vol. 9, pp. 75729-75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

41. N. Islam et al., Towards Machine Learning Based Intrusion Detection in IoT Networks," Comput. Mater. Contin., vol. 69, no. 2, pp. 1801-1821, 2021, doi: 10.32604/cmc.2021.018466.

42. A. Ayantayo et al., "Network intrusion detection using feature fusion with deep learning," J. Big Data, vol. 10, no. 1, 2023, doi: 10.1186/s40537-023-00834-0.

43. J. S. Kushwah, D. Gupta, A. Shrivastava, P. Ambily Pramitha, J. T. Abraham, and M. Lunagaria, 'Analysis and visualization of proxy caching using LRU, AVL tree and BST with supervised machine learning,' Materials Today: Proceedings, vol. 51, pp. 750-755, 2022, doi: 10.1016/J.MATPR.2021.06.224.

44. J. Singh Kushwah, A. Kumar, S. Patel, R. Soni, A. Gawande, and S. Gupta, 'Comparative study of regressor and classifier with decision tree using modern tools,' Materials Today: Proceedings, vol. 56, pp. 3571-3576, 2022, doi: 10.1016/J.MATPR.2021.11.635.

45. S. Pathak, V. Gupta, N. Malsa, A. Ghosh, and R. N. Shaw, 'Blockchain-Based Academic Certificate Verification System. A Review,' Lecture Notes in Electrical Engineering, vol. 914, 2022, doi: 10.1007/978-981-19-2980-9_42.

46. M. S. Ghuraiya, J. S. Kushwah, and K. Shrivastava, 'The Study of Swarm Intelligence Technique: A Review,' [PDF], n.d.