Pseudo-Image Representation for CNN-Based Intrusion Detection

Hassene Chaibi, and Amine Marref

Original scientific article

***Abstract*—Convolutional Neural Networks (CNNs) have demonstrated strong potential for Intrusion Detection Systems (IDSs); however, their performance is highly dependent on input data representation. This paper presents a lightweight pseudo-image transformation framework that converts network traffic into optimized two-dimensional layouts for CNN processing, focusing on data representation rather than architectural modifications. Evaluated on NSL-KDD and CIC-IDS2017 datasets, square and near-square formats are shown to enhance spatial feature extraction. In particular, an 11×11 base representation, upscaled to 64×64 using bicubic interpolation, achieves a precision of 99.2%, accelerates convergence, and remains robust under limited training data conditions. By avoiding complex hybrid architectures, the proposed approach reduces computational overhead while maintaining competitive performance, making it suitable for real-time, large-scale deployment. These results suggest that strategic data representation, rather than increased model complexity, can significantly improve CNN-based IDS performance.**

***Index Terms*—Convolutional Neural Networks (CNNs), Data Representation, Intrusion Detection Systems (IDSs), Pseudo-Images, Cybersecurity, NSL-KDD, CIC-IDS2017.**

I. Introduction

The rapid evolution of information and communication technologies has been accompanied by a corresponding increase in the scale, diversity, and sophistication of cyber threats. Conventional security mechanisms, which often rely on static rules or handcrafted signatures, are increasingly inadequate for protecting modern networks against complex and evolving attack patterns. As a result, data-driven approaches based on machine learning and deep learning have gained considerable attention in the design of Intrusion Detection Systems (IDSs).

Manuscript received April 29, 2026; revised May 20, 2026. Date of publication August 27, 2026. Date of current version August 27, 2026. The associate editor prof. Hrvoje Karna has been coordinating the review of this manuscript and approved it for publication.

H. Chaibi (corresponding author) is with the Computer Science Department, Faculty of Mathematics, Informatics, and Telecommunications, University of Saida - Dr. Moulay Tahar, Saida, Algeria and with Lab. LITIO, University of Oran 1 Ahmed Ben Bella, Algeria (e-mail: hassene.chaibi@univ-saida.dz).

A. Marref is with the Computer Science Department, Faculty of Mathematics, Informatics, and Telecommunications, University of Saida – Dr. Moulay Tahar, Saida, Algeria (e-mail: amine.marref@univ-saida.dz).

Digital Object Identifier (DOI): 10.24138/jcomss-2026-0085

Among these approaches, Convolutional Neural Networks (CNNs) have emerged as a promising solution due to their ability to automatically extract hierarchical and discriminative features. Originally developed for image recognition tasks, network traffic data into structured visual representations. This transformation enables CNNs to capture spatial correlations among traffic features that are difficult to model using traditional classifiers or purely sequential learning models.

Despite their potential, the performance and practicality of CNN-based IDSs are highly dependent on how non-image network traffic data are represented. Data representation is often treated as a secondary implementation detail, while research efforts predominantly focus on increasingly complex architectures or hybrid models. However, inappropriate or inefficient data transformations may introduce unnecessary computational overhead or fail to fully exploit the strengths of convolutional learning. Rather than proposing a new CNN architecture, this work systematically investigates how data representation alone influences CNN-based IDS performance.

In this work, we conduct a controlled and comparative study of data representation techniques for CNN-based intrusion detection. Several pseudo-image configurations are evaluated to analyse their impact on detection precision, learning stability, robustness under reduced training data, and execution time. Experiments are carried out on the NSL-KDD and CIC-IDS2017 benchmark datasets, allowing the proposed approach to be assessed under both classical and highly diverse network traffic scenarios.

The results demonstrate that carefully designed two-dimensional square and near-square representations can significantly enhance detection performance while maintaining computational efficiency. By identifying an optimized data transformation configuration, this study shows that competitive performance can be achieved without relying on overly deep or complex architectures, thereby improving the suitability of CNN-based IDSs for real-time or large-scale deployment.

The main contributions of this paper are summarized as follows:

1. A systematic investigation of data representation and transformation strategies for converting network traffic data into pseudo-images suitable for CNN-based intrusion detection.
2. A controlled comparative evaluation of multiple pseudo-image configurations, analysing their impact on detection

1845-6421/12/2026-0085 © 2026 CCIS

3. precision, robustness, learning dynamics, and execution time; while keeping the CNN architecture fixed across all experiments.
4. An empirical assessment of efficiency-performance trade-offs, highlighting how optimized data representation can reduce computational overhead while preserving high detection accuracy.
5. A structured comparison with classical and state-of-the-art IDS models, demonstrating that effective data transformation can achieve competitive results with reduced architectural complexity.

The remainder of this paper is organized as follows. Section II presents a comprehensive review of related work, focusing on CNN-based intrusion detection systems and data representation strategies. Section III describes the proposed methodology, including the pseudo-image generation process, data preprocessing, and CNN architecture. Section IV reports and discusses the experimental results obtained on the NSL-KDD and CIC-IDS2017 datasets, along with a comparative analysis against existing approaches. Finally, Section V concludes the paper and outlines potential directions for future research.

II. Literature Review

The application of Convolutional Neural Networks (CNNs) in Intrusion Detection Systems (IDSs) has gained significant attention due to their ability to learn hierarchical and discriminative features from network traffic data. Compared to traditional machine learning methods, CNN-based approaches achieve higher accuracy, robustness, and scalability in complex environments. However, their effectiveness strongly depends on how input data are represented. Recent studies such as [1], [2], and [3] show that performance improvements are often driven more by preprocessing and data representation than by architectural complexity, highlighting the critical role of input design.

Beyond CNNs, alternative paradigms such as Recurrent Neural Networks (RNNs), generative models, and Transformer-based approaches have been explored. Hybrid models combining CNNs with LSTM or BiLSTM effectively capture both spatial and temporal dependencies. For instance, [1] proposes an attention-based CNN-LSTM, while [4] extends this idea using federated learning for IoT IDSs. Transformer-based models [5] further process raw traffic without manual feature engineering. However, these approaches often introduce high computational cost and latency [6], limiting real-time applicability. Consequently, optimizing data representation emerges as a more efficient alternative to increasing architectural complexity.

A. Data Representation and Transformation for CNN-Based IDSs

A key challenge in CNN-based IDSs is transforming tabular data into formats suitable for convolutional processing. Early work [7] showed that reshaping feature vectors into 2D matrices enables CNNs to exploit spatial correlations. More advanced approaches, such as [8], combine dimensionality reduction (PCA, UMAP) and autoencoders to generate image-like representations. Although effective, these methods may introduce information loss and produce arbitrary spatial arrangements, limiting feature interpretability.

Not all transformations are equally effective. Structured representations that preserve spatial locality better support CNN learning. In contrast, augmentation-based methods such as GANs [9] and diffusion models [10], as well as multi-level IDS frameworks [11], primarily improve accuracy and class balance without optimizing spatial feature organization. Similarly, preprocessing techniques such as SMOTE-ENN [12] do not address representation limitations.

Alternative methods like 1D CNNs [13] avoid transformation by processing sequential data directly, offering efficiency but limited spatial feature learning. Lightweight CNNs [14], [3] reduce computational cost but may struggle with complex patterns if data representation is suboptimal.

Dataset characteristics also influence performance evaluation. NSL-KDD is relatively structured and easier to model, while CIC-IDS2017 is more realistic and complex. Therefore, robust representation strategies must generalize across datasets. Overall, these observations emphasize the importance of designing transformation methods that preserve spatial structure while minimizing information loss.

B. Architectural Enhancements and Hybrid Learning Approaches

To improve detection performance, many studies focus on architectural enhancements and hybrid models. Optimization-based methods [15] rely on metaheuristics for feature selection and tuning, while [16] propose a preprocessing framework combining Boruta feature selection and semi-supervised clustering to enhance IDS performance through noise reduction and feature relevance. In addition, [17] proposed SA-BO-CNN, integrating autoencoders, SMOTE, and Bayesian optimization. Although effective, these approaches introduce additional complexity.

Advanced hybrid models further integrate spatial and temporal learning. Wang et al. [18] proposed Res-TranBiLSTM, combining ResNet, Transformer, and BiLSTM to capture spatio-temporal dependencies, while Udas et al. [19] introduced SPIDER, integrating enhanced RNNs with PCA for robust detection.

Hybrid and ensemble approaches [2], [20] combine CNNs with classical classifiers to improve accuracy, though at the cost of increased complexity and reduced interpretability. Ebrahimi et al. [21] proposed a lightweight 1D-CNN with explainable AI (SHAP, LIME), improving interpretability for IoT environments but adding processing overhead.

More advanced architectures include CNN-BiLSTM models [6], which improve accuracy but increase latency. Multi-scale

CNNs [22] capture features at multiple resolutions, while channel attention CNNs [23] enhance feature importance. Transfer learning approaches [24] improve generalization and reduce training time. Despite their effectiveness, these methods often increase computational cost.

C. Deployment-Oriented CNN-Based IDSs

Practical IDS deployment requires a balance between accuracy and efficiency. Lightweight CNNs [14], [3] reduce model size and inference time, making them suitable for IoT environments, though sometimes at the expense of robustness.

Ensemble models [25] improve reliability but increase complexity, while federated learning approaches [4] address privacy but introduce communication overhead. Generative approaches, such as GAN-based IDSs in fog computing [9] and diffusion models [10], enhance detection in large-scale environments but require complex training and resources.

Dataset selection also impacts deployment evaluation. NSL-KDD remains widely used but lacks realism, whereas CIC-IDS2017 provides more representative traffic patterns. Therefore, cross-dataset evaluation is essential for real-world applicability.

D. Discussion and Research Gap

The literature shows that CNN-based IDSs achieve strong performance across datasets [26], [27], yet several limitations remain.

First, data representation is often underexplored despite its critical impact. Methods such as [8] may introduce information loss, and approaches like [28] focus on preprocessing without explicitly optimizing spatial organization. Even advanced models [22], [24] improve accuracy without addressing input structure.

Second, many studies report high performance on NSL-KDD but lack validation on more complex datasets like CIC-IDS2017. Generative methods [9], [10] enhance data diversity but do not optimize spatial feature arrangement.

Third, advanced architectures including Transformers [5] and hybrid models [1], [2], [6] improve detection but significantly increase computational complexity and latency, limiting real-time deployment.

Finally, traditional models such as Random Forest remain competitive due to their efficiency and low latency [6], making them important baselines.

To address these limitations, this study adopts a data-centric approach focusing on optimized data representation rather than architectural complexity. A pseudo-image transformation framework is proposed to preserve spatial coherence and minimize information loss. The approach is evaluated on NSL-KDD and CIC-IDS2017, and compared with both Random Forest and state-of-the-art CNN-based models. The objective is to demonstrate that optimized representation can achieve high precision, fast convergence, and reduced computational cost, offering an effective alternative to complex architectures.

III. METHODS

A. Pseudo-Image Generation

In this study, we explore the generation of pseudo-images for training CNNs in intrusion detection tasks. As CNNs are inherently designed for visual inputs, non-image network traffic data must be converted into structured image-like representations. The proposed transformation process involves three main steps: (1) normalization of feature values, (2) mapping one-dimensional feature vectors into two-dimensional frames, and (3) embedding and resizing these frames into square pseudo-images suitable for CNN processing.

By experimenting with different frame formats including square, vertical rectangular, and horizontal rectangular layouts, we analyze how various structural representations affect the CNN's ability to capture patterns and improve detection performance. It is important to note that network traffic features do not inherently possess spatial locality. The goal of this work is not to claim inherent spatial semantics, but to empirically investigate whether structured feature organization can facilitate convolutional feature extraction and stabilize learning dynamics.

A.1 Data Normalization:

In the context of a CNN-based IDSs, data normalization is a crucial preprocessing step aimed at harmonizing the scale of features extracted from network traffic. The raw attributes such as connection durations, packet counts, and flow statistics exhibit heterogeneous numerical ranges that may introduce bias into the loss function and hinder the learning process.

Let (x) denote a raw value associated with a network traffic feature. The normalization adopted in this study is defined as:

$$x_{norm} = \frac{x - X_{min}}{X_{max} - X_{min}} \times 255 \qquad (1)$$

where X_{min} and X_{max} respectively represent the minimum and maximum observed values of the feature within the dataset. This transformation projects all feature values into the interval [0,255], which corresponds to the pixel intensity range of conventional grayscale images.

Such scaling enables a consistent and meaningful pseudo-image representation of network traffic data, making it suitable for convolutional processing. Furthermore, it contributes to stabilizing gradient backpropagation, accelerating convergence during training, and improving the model's ability to discriminate between normal and malicious behaviors.

A.2 Transforming Data Vectors into Frames for CNN Processing:

In the proposed approach, each network traffic sample is initially represented as a one-dimensional vector of size T. This vector is then reshaped into a two-dimensional frame with

dimensions H×L, such that the constraint T = H × L is satisfied (Fig. 1).

This transformation allows the data to be represented in different visual configurations, namely:

1. Square Frame: when $H = L$, the data vector is reshaped into a square matrix.
2. Vertical Rectangular Frame: when $H > L$, the resulting frame has a vertical orientation, potentially emphasizing feature continuity along the height dimension.
3. Horizontal Rectangular Frame: when $L > H$, the frame is horizontally oriented, which may highlight correlations along the width dimension.

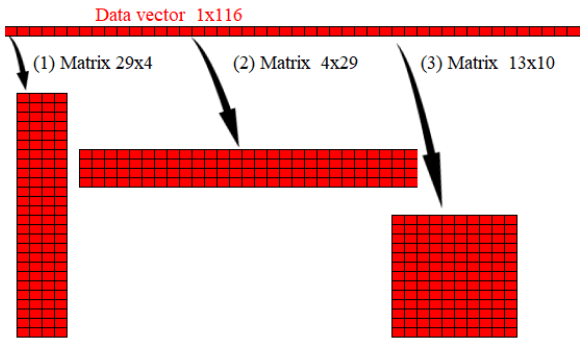


Fig. 1. Example of Data Vector Reshaping into a Matrix.

These alternative frame configurations enable the exploration of multiple spatial organizations of the same data, each potentially influencing the way convolutional filters extract local patterns. Several frame formats are evaluated in this study. The different configurations used are summarized in Table I, which presents the tested frame sizes.

TABLE I
DIFFERENT FRAME SIZES USED FOR DATA REPRESENTATION

Test	Frame size (H × L)
1	1 x 121 / 1x79
2	121 x 1 / 79x1
3	4x31 / 4 x 20
4	31x4 / 20 x 4
5	11 x 11
6	10 x 13
7	13 x 10

For the linear representation, two distinct configurations are employed, each adapted to the feature dimensionality of the respective dataset. Specifically, a 1×121 layout is used for NSL-KDD, while a 1×79 layout is applied to CIC-IDS2017 after preprocessing.

A.3 Embedding and Resizing Data as Square Pseudo-Images for CNN Input:

To ensure compatibility with standard CNN architectures typically designed for square image inputs the previously generated frames are embedded into square matrices to form pseudo-images. In this study, three target resolutions are considered: 32 × 32, 64 × 64, and 128 × 128.

Once embedded, each frame is resized to fill the entire image area (Fig. 2). Two resizing strategies are investigated:

- Non-resampling resizing, where the frame is directly scaled to the target resolution without applying interpolation.
- Interpolation-based resizing: which adapts pseudo-images to the fixed input size required by CNNs. In this approach, interpolation functions are applied to maintain spatial continuity between adjacent features, thereby reducing structural distortion during resizing. It is important to clarify that this "resampling" refers strictly to image interpolation for resolution adjustment and spatial smoothing. It does not alter the underlying class distribution nor does it function as a data balancing technique.

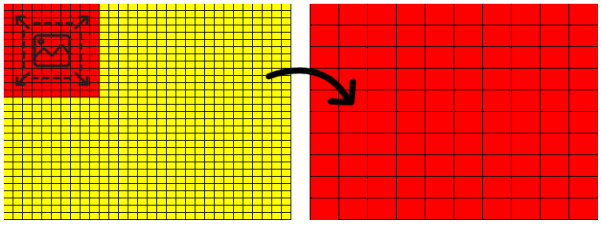


Fig. 2. Image resizing (from frame size 11x11 (red rectangular) to image size 32x32).

In this work, bicubic interpolation is employed for the resampling process. This technique estimates the intensity of each generated pixel using a cubic convolution over a 4 × 4 neighbourhood (16 surrounding pixels). The interpolated value is computed as a distance-weighted combination of these neighbouring pixels, where the weights are determined by a cubic polynomial function. Compared with simpler interpolation methods, bicubic interpolation produces smoother intensity transitions and better preserves structural coherence, making it particularly suitable for upscaling pseudo-images while maintaining spatial quality.

To illustrate the impact of the two resizing strategies, representative pseudo-images in the 64 × 64 format are presented. Fig. 3 shows examples generated without resampling (images a1–a3), whereas Fig. 4 illustrates the corresponding pseudo-images obtained with resampling enabled (images b1–b3).

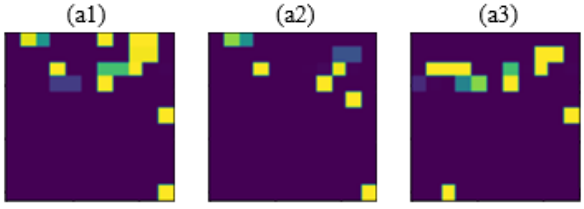


Fig. 3. Pseudo-images Generated from an 11×11 Frame (Image Size: 64×64, No Resampling).

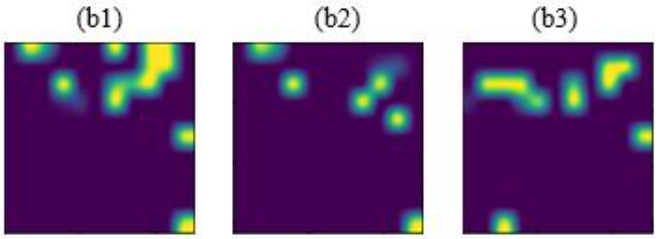


Fig. 4. Pseudo-images Generated from an 11×11 Frame (Image Size: 64×64, With Resampling).

B. Dataset Description

To evaluate the effectiveness of the proposed CNN-based intrusion detection framework, experiments are conducted on two widely used benchmark datasets: NSL-KDD and CIC-IDS2017. Although both datasets are originally composed of tabular network traffic features, their structured numerical and categorical attributes make them suitable for transformation into image-like representations. This transformation enables the exploitation of spatial feature correlations by CNNs. Both datasets are evaluated under a multi-class intrusion detection setting.

The NSL-KDD dataset is employed to assess the performance of the proposed model in detecting network intrusions within a controlled benchmark environment. It consists of 42 attributes, and addresses key limitations of the original KDD99 dataset by eliminating redundant records, leading to more reliable and unbiased classification results. The 42nd attribute corresponds to the class label, which indicates whether a given network connection is normal or malicious by specifying the attack type. Intrusion instances are categorized into four main classes: Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). The distribution of samples across these classes is reported in Table II. From a data representation perspective, the numerical and encoded categorical features of NSL-KDD are normalized and reorganized into fixed-size two-dimensional matrices, which are subsequently mapped to grayscale pseudo-images. This representation preserves the relative relationships between features and allows the CNN to capture local spatial patterns associated with different intrusion behaviors.

To further validate the robustness of the proposed approach under realistic and contemporary network conditions, the CIC-IDS2017 dataset is utilized. This dataset includes a wide range of recent attack scenarios derived from real-world network traffic captures (PCAP files). Network traffic flows are generated using the CICFlowMeter tool and labeled based on timestamps, source and destination IP addresses, ports, protocols, and attack types, resulting in structured CSV files.

The dataset contains 84 extracted network traffic features, with the final attribute representing the multi-class label. In addition, CIC-IDS2017 simulates the abstract behaviors of 25 users across common network services such as HTTP, HTTPS, FTP, SSH, and email, thereby reflecting realistic usage patterns. Compared to publicly available IDS datasets released between 1998 and 2016, CIC-IDS2017 satisfies 11 comprehensive performance evaluation criteria. The distribution of attack classes and their corresponding numbers of instances is presented in Table III.

Similar to NSL-KDD, the CIC-IDS2017 features are subjected to normalization and encoding prior to being reshaped into two-dimensional pseudo-images. Given the higher dimensionality of CIC-IDS2017, careful feature arrangement and image size selection are applied to ensure consistent input dimensions for the CNN. This transformation enables the convolutional layers to effectively learn discriminative spatial representations from complex network traffic patterns.

TABLE II
NUMBER OF DIFFERENT CLASSES IN NSL-KDD

Attack/Class	Normal connections	DoS	Probe	R2L	U2R
Number	77054	53385	14077	3749	252

TABLE III
NUMBER OF DIFFERENT CLASSES IN CIC-IDS2017

Attack /Class	Normal connection	DoS Hulk	DDoS	Port-Scan	DosGolden-Eye
Number	2096134	172846	128016	90819	10286

C. Test Protocol

To assess the effectiveness and reliability of the proposed approach, a structured testing protocol is designed and implemented. This protocol aims to systematically evaluate the quality and efficiency of the proposed CNN-based intrusion detection framework. It consists of a sequence of well-defined stages, beginning with data preprocessing and culminating in performance evaluation.

The testing protocol comprises the following steps:

1. Data Preparation: This initial step ensures that the raw network traffic data are properly preprocessed and formatted to be compatible with the proposed model. It includes data cleaning, normalization, encoding of categorical features, and label organization.
2. Pseudo-Image Generation: The second and most critical step focuses on transforming the preprocessed tabular feature vectors into two-dimensional pseudo-images. Various image sizes and layouts are considered to analyze their impact on the CNN's learning capability and detection performance.
3. CNN Configuration: In this step, only minor input-related adjustments are performed to ensure compatibility with different pseudo-image dimensions; the CNN architecture remains intentionally simple and largely unchanged throughout the study.

4. Training Phase: During this phase, the CNN model is trained using the generated pseudo-images. The learning process enables the model to automatically extract discriminative spatial features associated with normal and malicious network behaviors.
5. Testing Phase: Finally, the trained model is evaluated on unseen data to assess its generalization capability. Performance metrics are computed and compared across different experimental settings to validate the effectiveness of the proposed approach.

D. Performance Evaluation

The performance of the proposed intrusion detection approaches is assessed using a complementary set of metrics, selected according to the study objectives and the operational constraints of IDS. Precision serves as the primary evaluation metric, supplemented by per-class Recall and F1-score. This selection is motivated by the need to balance detection quality and operational feasibility: precision quantifies the proportion of correctly identified intrusions among all samples classified as attacks, a critical criterion for preventing alert flooding in production environments. However, given the severe class imbalance characteristic of cybersecurity datasets, recall and F1-score are essential for specifically evaluating the detection capability of minority classes.

Furthermore, execution time is employed as an additional criterion to assess the computational efficiency of the proposed models. This measure is particularly relevant in security contexts where detection must be performed in real time or under limited computational resources. The combination of precision and execution time thus enables a balanced evaluation of both detection quality and practical viability.

For comparative analysis across different learning models, the confusion matrix is utilized to provide a detailed decomposition of classification outcomes, facilitating a transparent analysis of detection behavior. The evaluation metrics are derived from the four fundamental components of this matrix: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Precision, Recall, and F1-score are formally defined as follows:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

E. CNN Model Architecture

The CNN architecture used in this work is intentionally conventional. The objective of this study is not to propose a novel CNN design, but rather to isolate and evaluate the impact of pseudo-image data representation on intrusion detection performance. The proposed architecture is summarized in Table IV.

ReLU activation is applied in hidden layers, and the model is trained using categorical cross-entropy with the Adam optimizer. Regularization is ensured through dropout and L2 weight decay to improve generalization. The datasets were first randomly shuffled using a fixed random seed to ensure reproducibility.

A stratified splitting strategy was employed to preserve class distributions across subsets. The default experimental configuration used 80% of the dataset for training and 20% for testing. During training, the training subset itself was further divided into 90% effective training ;10% validation

The validation subset was used exclusively for monitoring validation loss and early stopping. Final reported metrics were computed only on the independent test set.

The model is trained for 10 epochs with a batch size of 32 under a fixed random state for reproducibility.

TABLE IV
ARCHITECTURE OF THE CNN MODEL

Layer	Layer Type	Parameters / Configuration	Output Description
1	Input Layer	Pseudo-image of size (H × W × 1)	Grayscale input
2	Convolutional Layer	32 filters, (3×3), stride=1, padding = same	Feature maps
3	Activation	ReLU	Non-linearity
4	Max Pooling	(2 × 2)	Down-sampling
5	Convolutional Layer	64 filters, (3×3), stride = 1, padding = same	Feature maps
6	Activation	ReLU	Non-linearity
7	Max Pooling	(2 × 2)	Down-sampling
8	Dropout	(p = 0.2)	Regularization
9	Flatten	—	1D feature vector
10	Fully Connected	128 neurons	High-level features
11	Activation	ReLU	Non-linearity
12	Dropout	(p = 0.5)	Regularization
13	Output Layer	5 neurons, Softmax	Multi-class output

F. Data Preparation

- Attribute Removal: The first step involves removing features that are irrelevant for the CNN-based intrusion detection task.
- Class Distribution: The relationships among features are analyzed to categorize network events into standard classes: Normal, DoS, R2L, Probe, and U2R. This categorization ensures that the multi-class problem is well-structured for model training.
- Normalization: A data frame is constructed with the multi-class labels identified in the previous step. Numerical features are then scaled using a standard normalization approach (e.g., min-max scaling to [0,

255]) to harmonize the range of values. This normalization is crucial for representing tabular network data as pseudo-images, enabling the CNN to effectively capture patterns across all features.

- One-Hot Encoding of Categorical Variables: Categorical attributes, initially represented as strings, are converted into numerical vectors via one-hot encoding. This transformation allows the construction of a feature matrix suitable for reshaping into 2D pseudo-images, ensuring compatibility with the convolutional layers of the CNN.

IV. Results and Discussion

A series of experiments were conducted to evaluate the proposed CNN-based IDSs, considering datasets, training size, classifiers, and pseudo-image dimensions. The experimental protocol was divided into four phases: (1) image size variation, (2) training dataset size manipulation, (3) Performance Evaluation and Convergence Analysis of the Proposed CNN Model, (4) CNN versus Random Forest comparison, and (5) Comparative Analysis of CNN-based IDSs on NSL-KDD and CIC-IDS2017.

A. Image Size Variation

The first experiment investigates the impact of frame configurations and pseudo-image dimensions on detection performance and computational efficiency. Each configuration is denoted as $H{\times}L/D/R$, where $H{\times}L$ represents the frame size, D the image resolution (32, 64, 128), and R indicates whether resampling (interpolation) is applied.

The results obtained on the NSL-KDD (Fig. 5) and CIC-IDS2017 (Fig. 6) datasets, provide a comprehensive evaluation of the trade-off between precision and execution time.

In addition, a 1D-CNN baseline was implemented as a reference model, and its results are reported in the experimental evaluation figures. The 1D-CNN baseline consistently achieves lower precision compared to all 2D representation-based approaches, while exhibiting comparable execution times.

1) NSL-KDD Dataset: Without interpolation, all configurations achieve high precision (≥ 0.988). However, 1D layouts (1×121/128 and 121×1/128) show lower precision (≈0.9855–0.9856) and high execution times (5.42–5.95 s), confirming their limitations.

In contrast, 2D square and near-square layouts significantly improve performance. The best precision is obtained with 13×10/128 (0.9922, 5.95 s), while 11×11/64 (0.9916, 2.98 s) achieves similar precision with much lower cost. Lightweight configurations such as 10×13/32 (0.9902, 2.05 s) and 13×10/32 (0.9911, 2.06 s) provide an excellent trade-off for real-time use. Increasing depth from 32 to 128 only slightly improves precision but nearly doubles execution time.

With interpolation, precision improves slightly, especially for deeper models. The best results are achieved by 11×11/128 and 13×10/128 (0.9935) with execution times around 5.3 s. However, 11×11/64 (0.9924, 2.95 s) and 13×10/64 (0.9921, 2.98 s) offer a better balance. In some cases, interpolation also reduces runtime (e.g., 11×11/32: 2.65 s → 2.12 s), indicating more efficient learning.

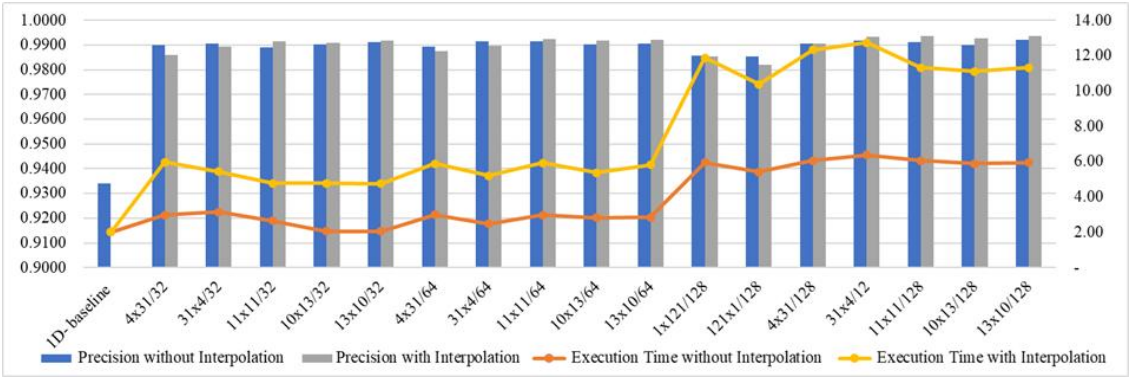


Fig. 5. Precision Comparison on the NSL-KDD Dataset.

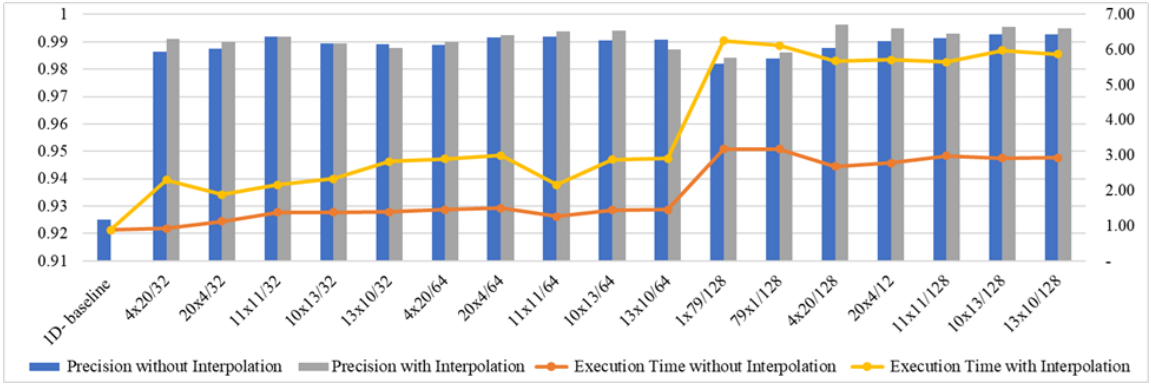


Fig. 6. Precision Comparison on the CIC-IDS2017 Dataset.

2) CIC-IDS2017 Dataset: Similar trends are observed, with generally lower execution times. Without interpolation, 2D configurations outperform 1D ones. The best result is 10×13/128 (0.9928, 2.92 s), while 1D layouts remain less effective (≈0.982–0.9838, ~3.18 s). Moderate configurations such as 11×11/32 (0.9918, 1.38 s) and 10×13/64 (0.9905, 1.44 s) already provide high precision with low cost.

With interpolation, performance improves significantly. The highest precision is achieved by 4×20/128 (0.9962, 3.00 s) and 10×13/128 (0.9954, 3.06 s). However, 11×11/64 (0.9937, 0.90 s) and 4×20/64 (0.9949, 0.88 s) demonstrate that moderate configurations can achieve excellent precision with very low execution time. In some cases, interpolation halves runtime (e.g., 11×11/64: 1.90 s → 0.90 s).

3) Key Observations Across both datasets:

- Superiority of 2D representations: Two-dimensional layouts significantly outperform 1D representations.
- Optimality of square and near-square layouts: Configurations such as 11×11, 10×13, and 13×10 provide the best balance.
- Impact of interpolation: Interpolation consistently improves detection performance by preserving spatial continuity, particularly for complex datasets.
- Efficiency of moderate sizes (32–64): These configurations achieve near-optimal precision with significantly lower execution times.
- Diminishing returns of large configurations (128): Although larger pseudo-images achieve marginally higher precision, the gain does not justify the substantial increase in execution time.

Statistical Note: The reported precision values represent single-run evaluations. Small variations (e.g., 0.9916 vs. 0.9922) should be interpreted cautiously, and that these differences primarily indicate performance trends rather than statistically validated superiority.

Overall, the results clearly demonstrate that data representation and computational efficiency are tightly coupled in CNN-based IDSs. Carefully designed pseudo-image structures particularly square or near-square layouts combined with moderate resolutions (64×64) offer the best trade-off between precision and execution time.

Interpolation further enhances this balance by preserving spatial continuity between neighbouring features and stabilizing learning, enabling convolutional filters to capture local correlations more effectively.

B. *Influence of Training Dataset Size*

In this test, the reference configuration 11×11/64/True (frame 11x11, image size = 64x64 and resampling is applied), identified in test 1, is used to evaluate the model's robustness against varying training ratios on the NSL-KDD and CIC-IDS2017 datasets. The experiments shown in Fig. 7 correspond to varying proportions of the original training subset (80%, 70%, 60%, and 50%) while maintaining the same independent test set. . For high training ratios (80% and 70%), both datasets achieve very high and stable precision, confirming the strong generalization capability of the selected configuration. As the training ratio decreases, a gradual drop in precision is observed, more pronounced for CIC-IDS2017 than for NSL-KDD, reflecting the higher complexity and heterogeneity of the network traffic. Nevertheless, even with reduced training ratios, the model maintains overall high performance, highlighting its robustness and the effectiveness of interpolation, combined with structured pseudo-image representation.

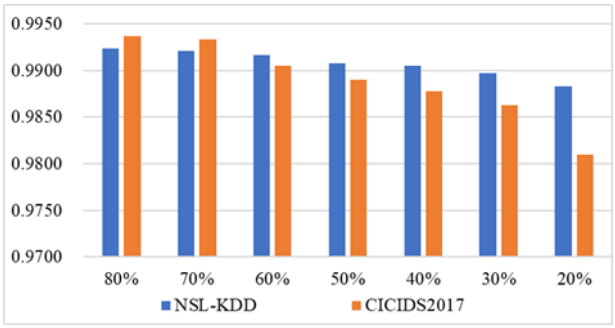


Fig. 7. Precision vs. training ratio for NSL-KDD and CIC-IDS2017.

C. *Performance Evaluation and Convergence Analysis of the Proposed CNN Model*

The proposed model was evaluated using the 11×11/64/True configuration over 10 epochs. Training was limited based on continuous monitoring of validation loss, which plateaued after 3–5 epochs with negligible precision gains (<0.2%) thereafter, justifying early termination. On NSL-KDD, the model rapidly achieved high precision with stable low loss. On CIC-IDS2017, convergence occurred within two epochs. These results confirm that the optimized pseudo-image representation enables efficient feature learning, eliminating the need for prolonged training.

D. *Comparison between CNN and Random Forest (RF)*

To further evaluate the performance of the proposed CNN model, a comparative analysis was conducted against the Random Forest (RF) algorithm. RF and CNN represent distinct algorithmic paradigms: RF is an ensemble of decision trees, where each tree independently classifies samples and contributes to the final prediction, while CNN leverages hierarchical feature extraction from pseudo-images.

The performance comparison is illustrated in the corresponding precision graphs for the NSL-KDD and CIC-IDS2017 datasets (Fig. 8). Across all training ratios, CNN consistently outperforms RF, demonstrating higher precision and more stable performance. For NSL-KDD, CNN maintains precision above 0.988 even with only 20% of the data used for training, whereas RF precision ranges from 0.869 to 0.913, showing greater sensitivity to training size. A similar trend is

observed on CIC-IDS2017, where CNN precision exceeds 0.981 across all ratios, while RF achieves 0.859–0.884, reflecting lower adaptability to high-dimensional and modern network traffic. These results highlight the superior capability of CNNs to extract discriminative features from pseudo-image representations, particularly under limited training data, whereas RF models exhibit notable performance degradation under the same conditions. Overall, the graphs confirm that the CNN-based approach provides a more robust and scalable solution for intrusion detection across diverse datasets.

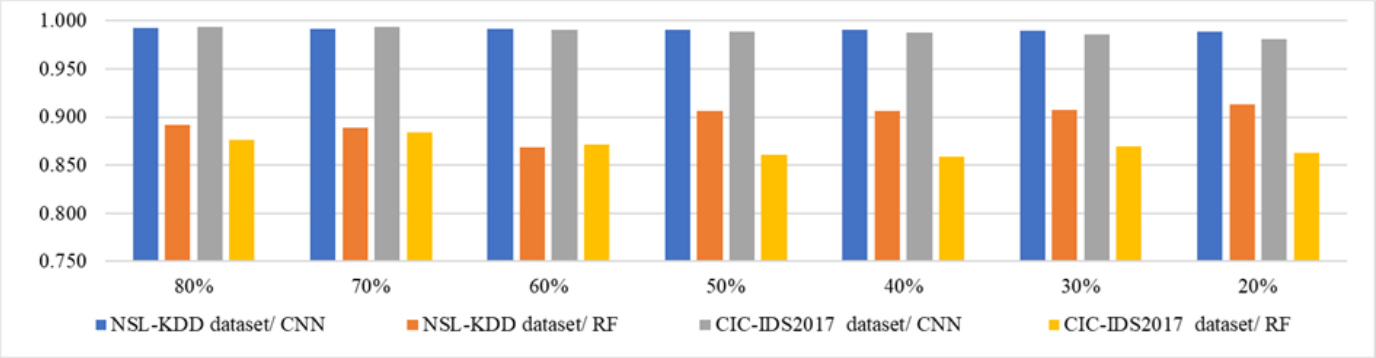


Fig. 8. Comparison between CNN and RF using NSL-KDD and CIC-IDS2017.

CNN Confusion Matrix

True Label	DoS	Probe	R2L	U2R	Normal
DoS	10663	7	0	0	7
Probe	3	2769	2	0	41
R2L	0	5	686	1	58
U2R	0	0	4	36	10
Normal	20	24	49	5	15313

Predicted Label

RF Confusion Matrix

True Label	DoS	Probe	R2L	U2R	Normal
DoS	8530	903	0	0	1244
Probe	33	2673	0	0	109
R2L	1	49	195	0	505
U2R	0	6	2	0	42
Normal	6	938	58	0	14409

Predicted Label

Fig. 9. Confusion matrix comparison at 80% training size (CNN vs. RF) on the NSL-KDD dataset.

CNN Confusion Matrix

True Label	BENIGN	GE-DoS	PortSc	DDoS	DoS
BENIGN	2000	2	10	7	4
GE-DoS	0	2070	0	0	0
PortScan	4	0	2033	2	0
DDoS	1	0	15	2090	0
DoS	2	0	0	0	2046

Predicted Label

RF Confusion Matrix

True Label	BENIGN	GE-DoS	PortSc	DDoS	DoS
BENIGN	2023	0	0	0	0
GE-DoS	487	1583	0	0	0
PortScan	656	0	1383	0	0
DDoS	181	0	0	1925	0
DoS	2038	0	0	0	10

Predicted Label

Fig. 10. Confusion matrix comparison at 80% training size (CNN vs. RF) on dataset CIC-IDS2017.

	dataset NSL-KDD					
	CNN results (%)			RF results (%)		
Classe	Recall	Precision	F1-score	Recall	Precision	F1-score
DoS	99.87	99.79	99.83	79.89	99.53	88.63
Probe	98.37	98.79	98.58	94.96	58.55	72.46
R2L	91.47	92.08	91.77	26	76.17	38.77
U2R	72	85.71	78.26	0	—	0
Normal	99.36	99.31	99.34	93.5	88.11	90.73

	dataset CIC-IDS2017					
	CNN results (%)			RF results (%)		
Classe	Recall	Precision	F1-Score	Recall	Precision	F1-Score
BENIGN	98.86	99.65	99.25	100	46.03	63.03
GE-DoS	100	99.9	99.95	76.52	100	86.7
PortScan	99.71	98.78	99.24	67.83	100	80.83
DDoS	99.24	99.57	99.4	91.41	100	95.5
DoS Hulk	99.9	99.8	99.85	0.49	100	0.97

Fig. 11. Performance comparison of CNN and RF in terms of Recall, Precision, and F1-Score with an 80% training split on the NSL-KDD and CIC-IDS2017 datasets.

The comparison between CNN and Random Forest (RF) reveals a clear advantage of the proposed CNN-based approach on both the NSL-KDD and CIC-IDS2017 datasets. As shown in the confusion matrices (Figs. 9 and 10), CNN produces significantly fewer misclassifications and achieves higher classification accuracy across nearly all attack categories. The performance metrics presented in Fig. 11 further confirm this superiority, with CNN consistently obtaining higher Recall, Precision, and F1-Score values than RF. The performance gap is particularly noticeable for minority classes such as R2L and

U2R in NSL-KDD and certain attack categories in CIC-IDS2017.

These differences can be partly attributed to the severe class imbalance present in both datasets. While RF tends to be biased toward majority classes, resulting in poor detection of rare attacks, CNN demonstrates greater robustness by maintaining high performance across both majority and minority classes. This suggests that the proposed pseudo-image representation, combined with convolutional feature extraction, enables CNN to capture more informative feature relationships and improve intrusion detection performance under imbalanced data conditions. Nevertheless, class imbalance remains a challenge and should be considered when interpreting the results.

E. Comparative Analysis of CNN-based IDSs

Beyond the Random Forest baseline, we further compared our approach with state-of-the-art deep learning-based IDS models reported in the literature.

All comparative results were standardized using the Precision metric and recomputed under identical evaluation settings to ensure a direct and unbiased performance assessment. The comparisons with existing literature are intended primarily for contextual benchmarking and should be interpreted with caution, as they may not be strictly comparable due to differences in preprocessing steps, dataset partitioning strategies, evaluation protocols, and hardware configurations.

E.1 Tests with NSL-KDD Dataset for multiclass classification:

The comparative results on the NSL-KDD dataset (Table V) highlight the progressive improvement in intrusion detection performance as models evolve from sequential and hybrid architectures toward spatially structured CNN-based approaches. Early models such as CNN-BiLSTM [6] and Hybrid RNN [19] achieve relatively modest precision (81.51% and 86.78%, respectively), while BiLSTM [10] improves performance to 93.1%, confirming the effectiveness of temporal modeling. Similarly, hybrid approaches, including Modified CNN + WCGAN-GP [8] (93.00%) and Hybrid Auto-encoder with CNN [26] (93.3%), demonstrate that combining multiple learning strategies enhances detection capability, albeit with increased architectural complexity.

In contrast, CNN-based models leveraging structured two-dimensional representations achieve significantly higher performance. While MCNN [28] attains moderate precision (84.00%), more advanced architectures such as multi-scale CNN [22] and CNN with channel attention [23] reach state-of-the-art results (99.67% and 99.7%, respectively). Notably, these models rely on square pseudo-image representations, emphasizing the importance of spatially coherent feature organization for effective convolutional learning.

Our proposed model achieves a precision of 99.24%, positioning it competitively among state-of-the-art CNN-based IDS solutions. Although slightly below the highest reported results, it offers a favorable trade-off between accuracy and complexity by combining a simple CNN architecture with an optimized 11×11 square data representation. This confirms that well-designed data transformation can significantly enhance detection performance without requiring complex architectural designs.

TABLE V
RESULTS OF CNN-BASED IDS MODELS WITH NSL-KDD DATASET

Models	Methods	Precision (%)
[6]	CNN-BiLSTM	81.51
[28]	MCNN	84.00
[19]	Hybrid RNN	86.78
[8]	Modified CNN + WCGAN-GP	93.00
[10]	BiLSTM	93.1
[26]	Hybrid Auto-encoder with CNN	93.3
[22]	Multi-scale CNN 13×13x3 / Color square format	99.67
[23]	CNN with channel attention 12 × 12 / 2D square format	99.7
Our model	Simple CNN 11x11	99.24

E.2 Tests with CIC-IDS2017 Dataset for multiclass classification:

The comparative results on the CIC-IDS2017 dataset (Table VI) further highlight the impact of model design under more complex and heterogeneous network conditions. Models based on generative and attention mechanisms, such as BiGAN [9] and Transformer-based IDS [5], achieve moderate precision (76.50% and 86.30%, respectively), indicating that these approaches alone are insufficient to effectively model complex intrusion patterns without appropriate feature structuring.

More advanced architectures demonstrate substantial performance improvements. The ResNet+TranBiLSTM model [18] achieves a high precision of 99.15%, illustrating the benefit of combining convolutional feature extraction with temporal dependency modeling. Similarly, PNet-IDS [3] further improves detection performance to 99.53%, confirming the effectiveness of deeper and more specialized architectures.

The best performance is obtained by the multi-scale CNN model [22] (99.84%), which relies on a structured 2D square pseudo-image representation. This result is consistent with the findings on NSL-KDD and reinforces the importance of spatially organized data representations for maximizing CNN effectiveness.

Our proposed model achieves a precision of 99.37%, demonstrating strong generalization capability across datasets. Despite its architectural simplicity, it delivers competitive performance, confirming that optimized pseudo-image representation plays a key role in enhancing detection accuracy. This balance between efficiency and performance makes the proposed approach particularly suitable for real-world and resource-constrained IDS deployment.

TABLE VI
RESULTS IDS BASED CNN WITH CIC-IDS2017 DATASET

Models	Methods	Precision (%)
[9]	BiGAN	76.50
[5]	IDSs with Transformers	86.30
[18]	ResNet+TranBiLSTM 28 × 28 / 2D square format	99.15
[3]	PNet-IDS	99.53
[22]	Multi-scale CNN 13 × 13 × 3 / 2D square format	99.84
Our model	Simple CNN 11x11	99.37

F. Discussion

The experimental results highlight the critical role of data representation in CNN-based intrusion detection. Two-dimensional pseudo-image layouts, particularly the 11×11/64 configuration, significantly outperform one-dimensional representations by enabling effective local feature learning while maintaining computational efficiency. The configuration 11×11/64/True indicates that preserving structural coherence while enhancing resolution via interpolation improves the CNN's ability to capture discriminative patterns.

The proposed model demonstrates fast convergence, high stability, and strong robustness across both datasets, with minimal performance degradation under reduced training data. Compared to Random Forest, the CNN consistently achieves higher precision, especially for complex and minority attack classes. Furthermore, comparisons with state-of-the-art IDS models show that the proposed approach achieves competitive performance despite its simpler architecture. These findings indicate that optimized data transformation, rather than increased model complexity, is a key factor in improving IDS performance.

Experimental Limitation. The reported results are based on a single run for each experimental configuration using a fixed random seed. Therefore, small performance differences between competing pseudo-image layouts should be interpreted with caution, as they indicate general performance trends rather than statistically validated superiority. Future work will include multiple independent runs and statistical analyses to further validate these findings.

Overall, this study demonstrates that carefully designed pseudo-image representation and interpolation provide an effective balance between accuracy, robustness, and computational cost. The results position data representation as a primary design dimension for developing efficient, scalable, and deployable CNN-based IDSs.

V. CONCLUSION

This study investigated the effectiveness of data representation techniques for CNN-based IDSs by systematically transforming tabular network traffic into structured pseudo-images. Extensive evaluation on the NSL-KDD and CIC-IDS2017 datasets, together with comparisons against recent CNN, RNN, hybrid, and attention-based IDS models, demonstrates that 2D square representations consistently improve detection precision, computational efficiency, and cross-dataset stability. The proposed CNN model achieves near state-of-the-art performance across datasets with heterogeneous traffic characteristics, highlighting its strong generalization capability and robustness to data variability. These results confirm that data representation constitutes a primary design dimension for CNN-based IDSs, often outweighing architectural complexity.

Future research will focus on automated feature layout optimization, repeated experiments with multiple random initializations to enable statistical significance analysis, advanced imbalance mitigation strategies, and direct comparison with additional sequential and tabular-learning baselines. Such experiments will provide stronger statistical evidence regarding the robustness of the proposed pseudo-image representations.

REFERENCES

[1] P. Phalaagae, A. M. Zungeru, A. Yahya, B. Sigweni and S. Rajalakshmi, "A Hybrid CNN-LSTM Model With Attention Mechanism for Improved Intrusion Detection in Wireless IoT Sensor Networks," IEEE Access, vol. 13, pp. 57322–57341, 2025, doi: 10.1109/ACCESS.2025.3555861.

[2] S. Khaitan and I. M. Meerasha, "AE-CNN Ensemble: A Novel Architecture for Effective Network Intrusion Detection and Classification," IEEE Access, vol. 14, pp. 1320–1340, 2026, doi: 10.1109/ACCESS.2025.3649245.

[3] A. S. Iliyasu, A. J. Siddiqui, H. Song and F. J. Abdu, "PNet-IDS: A Lightweight and Generalizable Convolutional Neural Network for Intrusion Detection in Internet of Things," IEEE Access, vol. 13, pp. 102624–102639, 2025, doi: 10.1109/ACCESS.2025.3575705.

[4] V. Prakash and A. K. Shukla, "An AI-Enabled Privacy-Preserving Federated Learning Framework of Hybrid bi-LSTM-RNN-CNN for Deep Intelligent Intrusion Detection in Fog Computing in the IoT Domain," Concurrency Computat. Pract. Exper., vol. 37, no. 27–28, p. e70291, 2025, doi.org/10.1002/cpe.70291

[5] K. Miyamoto, C. Han, T. Ban, T. Takahashi, and J. Takeuchi, "Intrusion detection simplified: A feature-free approach to traffic classification using transformers," in 2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops), pp. 20–29, 2024, doi: 10.1109/ACSACW65225.2024.00011.

[6] V. R. Joshi, K. Assa-Agyei, T. Al-Hadhrami and S. N. Qasem, "Hybrid AI Intrusion Detection: Balancing Accuracy and Efficiency," Sensors, vol. 25, no. 24, p. 7564, 2025, doi.org/10.3390/s25247564

[7] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," IEEE Access, vol. 7, pp. 42210–42219, 2019, doi: 10.1109/ACCESS.2019.2904620.

[8] P. Kumari, A. Srivastava and D. Sinha, "Converting Tabular to Image Data to Design IDS Applying Deep Learning Techniques," Innov. Syst. Softw. Eng., vol. 21, pp. 1543–1557, 2025, doi.org/10.1007/s11334-025-00619-z

[9] W. Yao, H. Shi, and H. Zhao, "Scalable anomaly-based intrusion detection for secure Internet of Things using generative adversarial networks in fog environment," Journal of Network and Computer Applications, vol. 214, Art. 103622, 2023, doi: 10.1016/j.jnca.2023.103622.

[10] B. Tang, Y. Lu, Q. Li, Y. Bai, J. Yu, and X. Yu, "A diffusion model based on network intrusion detection method for industrial cyber-physical systems," Sensors, vol. 23, no. 3, p. 1141, 2023, doi: 10.3390/s23031141.

[11] Mallidi, S.K.R., Ramisetty, R.R. A multi-level intrusion detection system for industrial IoT using bowerbird courtship-inspired feature selection and hybrid data balancing. Discov Computing 28, 109 (2025); doi.org/10.1007/s10791-025-09632-z.

[12] X. Zhang, J. Ran, and J. Mi, "An intrusion detection system based on convolutional neural network for imbalanced network traffic," in 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), pp. 456–460, 2019, doi: 10.1109/ICCSNT47585.2019.8962490.
[13] M. Arslan, M. Mubeen, M. Bilal, and S. F. Abbasi, "1D-CNN-IDS: 1D CNN-based intrusion detection system for IIoT," arXiv preprint arXiv:2401.17858, 2024, doi: 10.48550/arXiv.2401.17858.
[14] T.-T.-H. Le, A. A. Adiputra, A. A. N. Dharmawangsa, H. Jang and H. Kim, "Lightweight CNN-Based Intrusion Detection for CAN Bus Networks," IEEE Access, vol. 14, pp. 14870–14891, 2026, doi: 10.1109/ACCESS.2026.3654521.
[15] E. B. Elsayed, A. S. Yassin and H. Fahmy, "A Novel Hybrid GWO-RFO Metaheuristic Algorithm for Optimizing 1D-CNN Hyperparameters in IoT Intrusion Detection Systems," Information, vol. 16, no. 12, p. 1103, 2025, doi.org/10.3390/info16121103
[16] A. I. Harsapranata, E. Sediyono, and H. D. Purnomo, "Intrusion Detection: Boruta Feature Selection and Semi-Supervised Outlier Clustering with Multi-Dataset Evaluation", Eng. Technol. Appl. Sci. Res., vol. 15, no. 6, pp. 30283–30289, Dec. 2025, doi.org/10.48084/etasr.14351
[17] Y. Mo, H. Li, D. Wang, and G. Liu, "An intrusion detection system based on convolution neural network," PeerJ Computer Science, vol. 10, Art. e2152, 2024, doi: 10.7717/peerj-cs.2152.
[18] S. Wang, W. Xu, and Y. Liu, "Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things," Computer Networks, vol. 235, Art. 109982, 2023, doi: 10.1016/j.comnet.2023.109982.
[19] P. B. Udas, M. E. Karim, and K. S. Roy, "SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, part B, pp. 10246–10272, 2022, doi: 10.1016/j.jksuci.2022.10.019.
[20] T. Abid, A. Ahmim, F. Maazouzi et al., "A Novel IoT Threat Detection Using GWO Feature Selection and CNN-Enhanced LightGBM," J. Cloud Comput., vol. 14, p. 72, 2025, doi.org/10.1186/s13677-025-00785
[21] F. Ebrahimi, R. Javidan, R. Akbari et al., "Intrusion Detection in the Internet of Things Using Convolutional Neural Networks: An Explainable AI Approach," Cybersecurity, vol. 8, p. 66, 2025, doi.org/10.1186/s42400-025-00369-2
[22] H. Yang, J. Yu, and R. Zhai, "High-precision intrusion detection for cybersecurity communications based on multi-scale convolutional neural networks," The Journal of Supercomputing, vol. 81, no. 1, pp. 277–298, 2025, doi: 10.1007/s11227-024-06737-y.
[23] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and J. S. Alqurni, "CNN channel attention intrusion detection system using NSL-KDD dataset," Computers, Materials & Continua, vol. 79, no. 3, pp. 4319–4347, 2024, doi: 10.32604/cmc.2024.050586.
[24] F. Yan, G. Zhang, D. Zhang, X. Sun, B. Hou, and N. Yu, "TL-CNN-IDS: Transfer learning-based intrusion detection system using convolutional neural network," The Journal of Supercomputing, vol. 79, no. 15, pp. 17562–17584, 2023, doi: 10.1007/s11227-023-05347-4.
[25] M. Vishwakarma and N. Kesswani, "StaEn-IDS: An Explainable Stacking Ensemble Deep Neural Network-Based Intrusion Detection System for IoT," IEEE Access, vol. 13, pp. 109713–109728, 2025, doi: 10.1109/ACCESS.2025.3582391.
[26] L. Mohammadpour, T. C. Ling, A. Raza, and A. Aryanfar, "A survey of CNN-based network intrusion detection," Applied Sciences, vol. 12, no. 16, p. 8162, 2022, doi: 10.3390/app12168162.
[27] S. Sapre, P. Ahmadi, and K. Islam, "A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms," arXiv, 2019. Available: https://arxiv.org/abs/1912.13204.
[28] I. Al-Turaiki and N. Altwaijry, "A convolutional neural network for improved anomaly-based network intrusion detection," Big Data, vol. 9, no. 3, pp. 233–252, 2021, doi: 10.1089/big.2020.0263.



Hassene Chaibi was Head of the Information Technology Department at OAIC Saida from 1997 to 2011. Since 2011, he has been a professor at the University of Saida – Dr. Moulay Tahar, Algeria. His research interests include artificial intelligence, cybersecurity, image processing, and software engineering, with a focus on intelligent systems and real-world AI applications.



Amine Marref received the M.Eng. degree in Computer Systems and Software Engineering and the Ph.D. degree in Computer Science from the University of York, UK, in 2005 and 2009, respectively. From 2009 to 2010, he was a Postdoctoral Research Fellow at Mälardalen University, Sweden, supported by a fellowship from the Swedish Foundation for Strategic Research, where he worked on measurement-based worst-case execution-time (WCET) analysis for component-based real-time systems. From 2010 to 2014, he was an Assistant Professor at Umm Al-Qura University, Makkah, Saudi Arabia, where he led several nationally funded research projects on WCET analysis of parallel real-time systems and served as departmental e-learning director. Since 2023, he has been an Associate Professor at the University of Saida, Algeria. He is also the CEO and Founder of Infinity Computer Systems, where he develops the PrediWCET toolchain for WCET analysis and the AmiMaths suite of e-learning applications. His current research interests include machine learning for the timing analysis of real-time systems.