PREMIER JOURNAL OF SCIENCE ORIGINAL RESEARCH

An Intelligent IDS Using Feature Engineering and Fuzzy SVM-Based CNN: An Experimental Study

S. G. Suma and P. Rukmani

School of Computer Science and Engineering , Vellore Institute of Technology, Chennai, Tamil Nadu, India

Correspondence to:
S. G. Suma,
sumas.g2016@vitstudent.ac.in

Additional material is published online only. To view please visit the journal online.

Cite this as: Suma SG and Rukmani P. An Intelligent IDS Using Feature Engineering and Fuzzy SVM-Based CNN: An Experimental Study. Premier Journal of Science 2025;15:100224

DOI: https://doi.org/10.70389/PJS.100224

Peer Review

Received: 14 August 2025

Last revised: 22 October 2025

Accepted: 17 December 2025

Version accepted: 3

Published: 31 January 2026

Ethical approval: N/a

Consent: N/a

Funding: No industry funding

Conflicts of interest: N/a

Author contribution:
S. G. Suma and P. Rukmani – Conceptualization, Writing – original draft, review and editing

Guarantor: S. G. Suma

Provenance and peer-review: Unsolicited and externally peer-reviewed

Data availability statement: N/a

ABSTRACT

Recently, Machine Learning (ML) techniques have been applied in various Intrusion Detection Systems (IDS) to automatically detect and categorize known and unknown attacks. Various issues arise owing to the rapid change in the attack behavior in a large volume of data. Existing IDSs developed using ML algorithms often struggle to identify and detect various unknown attacks and fail to achieve the required detection accuracy for known attacks owing to a lack of learning and pattern identification. For this purpose, Deep Learning (DL) techniques are incorporated with IDSs to learn the dataset in depth, and identify the most important features, and achieve reasonable performance in terms of detection accuracy. However, these types of systems require a significant amount of time to predict attacks. An improved IDS framework is proposed in this study that applies feature engineering along with a hybrid Fuzzy Support Vector Machine (FSVM)-based Convolutional Neural Network (CNN) classifier to increase the detection rate and efficiency. Fuzzy logic enables FSVM to deal with uncertain data and it adds adaptive membership levels to inputs which produces better results than traditional SVMs while reducing the impact of ambiguous or noisy data samples. The refined data from the FSVM are processed by a CNN that identifies temporal attack patterns to enhance feature extraction accuracy and classification detection. By integrating FSVM with CNN, more precise attack detection is possible because imbalanced datasets are handled more effectively with greater generalization to new attack patterns. Using fuzzy logic and temporal constraints, the proposed IDS categorizes the network traffic accordingly into the modern attack families present in CICIDS2017 (DoS/DDoS, Brute Force, Infiltration, Web Attacks, Botnet, and Normal) and UNSW-NB15 (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, and Normal). We evaluated our approach using an extensive set of experiments on the CICIDS2017 and UNSW-NB15 benchmark datasets, and it showed better results in terms of detection accuracy and computational efficiency than existing ML and DL-based IDS models.

Keywords: CRF-IGR feature engineering, DOS, Fuzzy SVM–CNN hybrid classifier, Multi-class attack categorization (probe, NSL-KDD & CICIDS2017 benchmarking, R2L), Temporal traffic pattern analysis, U2R

Introduction

Information and Communications Technology (ICT) systems manage various types of user data, which are vulnerable to multiple forms of threats, including internal and external attacks. An Intrusion Detection System (IDS) is a software that is helpful for automatically detecting and classifying attacks in networks and Internet scenarios. Feature selection is helpful in improving classification accuracy, thereby reducing the time required for decision-making compared to using the full-featured dataset. The feature selection and extraction process is performed using feature engineering which performs the selection process through the information gain value of features and the various weights of features. The feature engineering process can be developed using a filter-based feature selection method and the respective statistical formulae. This study introduces a new feature engineering method that combines the Information Gain Ratio (IGR) and Conditional Random Field (CRF) approach for feature selection.

Machine Learning (ML) algorithms such as Neural Network (NN), Random Forest (RF), Naïve Bayes (NB), Decision Tree (DT), and Support Vector Machines (SVM), are widely used to classify and identify attack patterns. Many advanced versions of neural network algorithms, such as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Recurrent Neural Network (RNN), and Deep Belief Network (DBN), are available in the literature for predicting attack patterns. However, these models struggle to identify previously unobserved attacks. An improved IDS framework is proposed in this study that employs an anomaly detection strategy to monitor known and unknown attacks using hybrid Fuzzy SVM-CNN classifiers. The training stage works with labeled traffic data, which include standard operational patterns and known operational-attack counterparts. The system alerts potential anomalies whenever real-time deployment detects substantial deviations from the learned baseline behavior, which consists of irregular traffic patterns, uncommon communication protocols, and destination-source relationships. Anomaly detection using this method enables the system to recognize fresh attack varieties that exceed normal behavioral boundaries. Moreover, to handle uncertainty in network traffic, fuzzy logic is employed within SVM, whereas CNN optimizes the detection by learning the spatiotemporal pattern in the selected features. The two-layered classification architecture strengthens threat detection capabilities, particularly for sophisticated and changing security threats. The primary contributions of this study are as follows:

1. To develop a novel IDS that uses effective feature engineering and a Deep Learning (DL) model for effective intrusion detection.
2. We propose a new DL technique that combines the Fuzzy Support Vector Machine (FSVM) and CNN for effective classification.

3. To apply a newly proposed feature engineering technique that uses IGR and CRF to choose the most contributing features for efficient classification and to apply fuzzy logic to enhance decision-making on intrusion record sets.
4. To categorize network traffic into dataset-defined attack families of CICIDS2017 and UNSW-NB15 using a fuzzy classification mechanism.

Literature Survey

Numerous studies have focused on IDS, with particular emphasis on algorithms for feature selection and classification. In this area, ML and DL approaches have been widely used. A model that identified and selected the optimal number of features for finding intruders in an 802.11 network was developed in El-Khatib.[1] A feature selection algorithm for measuring the relevance between features using the information gain value of features was used. They demonstrated the significance and effectiveness of feature optimization in reducing the decision-making time for neural classifiers while achieving a higher detection accuracy than other methods. A new IDS using genetic network-based programming and fuzzy rule-based association mining was introduced in Mabu et al.[2] Adaboost-aware algorithms for effective intruder detection were developed in Hu et al.[3] A time-series model was developed in Vinayakumar et al.[4] to predict bad network connections using a Multilayer Perceptron (MLP), LSTM, and convolutional neural network-gated recurrent unit refinement unit (CNN-GURU) by considering the transmission control protocol packets and Internet protocol packets. An IDS using ML algorithms, such as Support Vector Classifier and MLP was proposed in Sadioura et al.[5] A new feed-forward deep neural network-based IDS incorporates a filter-based feature selection algorithm for predicting attacks and is compared with standard ML algorithms such as SVM, DT, RF, NB and K-Nearest Neighbors (K-NN) classifiers.[6] A decision tree-based IDS was built with the incorporation of a soft root sign activation method to detect attacks, including web attacks, brute force, DDoS, and infiltration.[7]

Dutt et al.[8] explored an immunological method for monitoring network traffic and detecting attacks. Moreover, they used inmate immune system and adaptive immune-based anomaly detection methods to identify suspicious activities in the network and relevant users. Lopez-Martin et al.[9] developed a new reinforcement learning method for performing supervised learning on network traffic datasets and proved that the model achieved better accuracy on the NSL-KDD dataset. A Particle Swarm Optimization (PSO) and hyperparameter-based feature subset method were developed in Elmasry et al.[10] to select the optimal features that are supportive for making actual decisions on network traffic datasets. A new feed-forward neural network incorporated the DL technique along with wrapper-based feature extraction that applies the extra trees method to generate optimal features.[11] Their technique performed better than the standard ML algorithms on the UNSW-NB 15 dataset. A DL that follows the single-dimensional approach on CNN for predicting known and unknown attacks effectively achieved better detection accuracy.[12] A CNN-based dimensionality reduction method was used to perform an effective intrusion detection process in Manikandan et al.[13]

A double-layer hybrid method for predicting different attacks, such as Probe, R2L, DoS and U2R, using Principal Component Analysis (PCA) and SVM on the NSL-KDD dataset was implemented in Wisanwanichthan et al.,[14] which achieved 100% accuracy in the detection of U2R attacks. A DL algorithm that incorporated an IDS that applied a pre-training method along with a deep autoencoder for performing the intrusion detection process was developed in Kunang et al.[15] As highlighted in Fatani et al.,[16] an Artificial Intelligence (AI)-based IDS that incorporates a CNN-aware feature extraction algorithm to extract more relevant features and a new optimization technique called differential evolution-based transient search optimization technique for enhancing the decision-making process on KDD, NSL-KDD, and CICIDS-2017 datasets was proposed, achieving superior accuracy than existing models.

A novel CNN-based IDS was introduced to enhance security in Internet communication.[17] They identified and detected intrusions by applying a CNN. Moreover, they used the CICIDS2017 dataset for training and testing and proved that the accuracy, overhead, and false-positive rate were better than those of existing models. A federated DL method for detecting cyber-attacks was designed in Li et al.[18] The design included a learning framework that builds a model for detecting attacks while preserving privacy, demonstrating improved accuracy.

A method that combines features based on time, byte level, and statistical attributes to extract meaningful data from various perspectives, enabling the development of an efficient model.[19] They introduced a new loss function for handling data imbalance. Their method enhances performance, and their loss function adjusts the weights. A novel intrusion detection approach was introduced, utilizing a CNN framework alongside a new overlapping method and optimized hyperparameter configuration.[20] Finally, their method achieved approximately 5% higher detection accuracy than other IDSs on the NSL-KDD dataset. Although Zhao et al.[21] used Generative Adversarial Network (GAN) based augmentation, Xi et al.[22] suggested a multi-scale transformer IDS, and Adjewa et al.[23] proposed a federated based IDS, the proposed FSVM-CNN is superior to them because it has higher accuracy, reduced false alarms and stronger generalization across benchmark datasets.

Although a great amount of research has been conducted on IDS, existing models are subject to issues such as high false positive rates, high computational inefficiencies, difficult adaptability to the evolution of attack strategies, and low real-time performance. To overcome these limitations, this study develops an enhanced IDS featuring an enhanced ML model incorporating a novel feature engineering approach in conjunction with a hybrid rule refinement model of FSVM and CNN with temporal learning capability. FSVM helps to reduce the noise and uncertainty in feature selection so that the classification accuracy improves, whereas CNN entails the sequential patterns in the network traffic, and when it is attacked, they are refined. The proposed IDS tackles the main challenges eluded in previous studies and ensures a higher detection rate, reduced false positives, and increased accuracy, making it suitable for solving modern cybersecurity problems.

System Architecture

The proposed IDS is composed of the following primary components, including dataset, user interaction module, Grouping Module, Decision Manager, Feature Selection Module, Classification Module, Knowledge Base, Rule Manager, and Fuzzifier as depicted in Figure 1.

The required data are gathered from datasets through the user interaction module, which is then sent it to the grouping module. This module organizes the features based on the values associated with the attack categories, such as DoS/DDoS, Brute Force, Infiltration, Web Attacks, Botnet, and Normal (CICIDS2017), as well as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, and Normal (UNSW-NB15). Once grouped, the features can be provided to the decision manager. The decision manager subsequently forwards the data to the feature selection module, where the most impactful features are identified using an efficient feature-engineering process. The selected featured record sets are then passed to the classification module by the decision manager, which performs effective classification using the double-layer approach.

In this approach, the FSVM handles the initial stage of classification, whereas the CNN performs a deeper analysis to capture temporal and spatial attack patterns. The final decisions on the various record sets are made by the decision manager with the help of a fuzzifier and knowledge base rules.

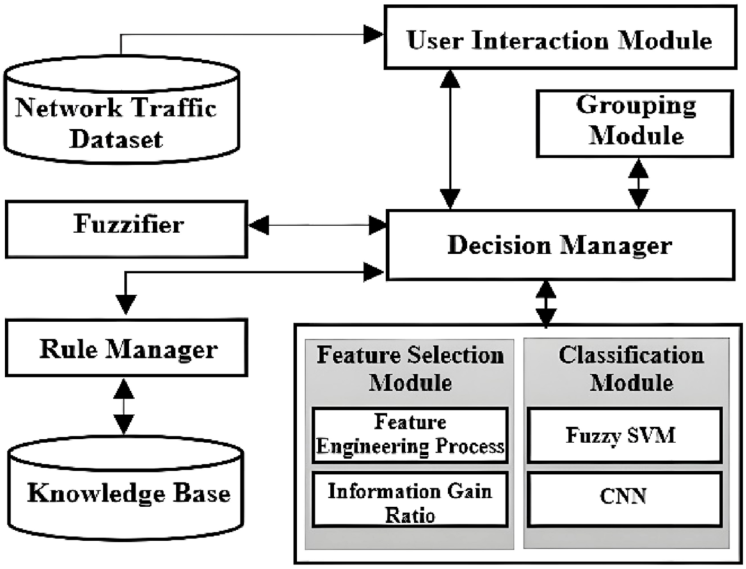

Fig 1 | Overall system architecture of the proposed fuzzy SVM-CNN IDS

Methods

This section illustrates the methods used in the proposed work, which mainly comprise feature engineering and classification as detailed in Algorithms 1–3. The feature selection stage utilizes newly proposed feature engineering methods that combine IGR, data normalization and CRF rules. Second, the classification process is performed using the proposed classifier that integrates Fuzzy SVM and CNN with temporal features to make dynamic and efficient decisions on record sets.

The feature selection process reduces the dimensions of network traffic dataset, thereby reducing the time required to analyze and classify record sets. The chosen features were helpful for improving the attack prediction and detection accuracy. This study introduces a novel feature engineering method aimed at achieving efficient feature selection. The performance of an IDS is largely influenced by the selected feature vector. Therefore, it is crucial to select relevant features for accurate classification. Intrusion detection data typically consist of traffic logs collected from various user devices and applications that often have inconsistent formats. Standardizing these formats helps the ML model learn more effectively. Furthermore, normalizing the logs can enhance the detection of similar patterns, aiding in the identification of anomalies. By reducing inconsistencies in the dataset through normalization, the number of false alarms can be significantly decreased. Since the intrusion dataset contains multiple features, min-max scaling was applied to ensure that all features were within the same range.[24]

Feature Engineering Method

The feature engineering process converts the raw packet inputs into normalized feature vectors, ranks the features using IGR and CRF and returns a reduced feature set F_{ranked} for classification. The feature vector $F = \{f_1, f_2, \ldots ,f_n\}$, where $1 < n < T$, with T denoting the total number of features and C representing the class label in the dataset, is processed by the feature engineering stage. The detailed procedure adopted in this process is outlined in Algorithm 1.

The CRF parameters were estimated using the L-BFGS optimizer with an L2 regularization factor $\lambda = 0.1$ and a convergence tolerance of 10^{-5}. The final selection used K = 32 features for CICIDS2017 and K = 27 for UNSW-NB15, chosen to balance accuracy and computational efficiency.

The CRF model was selected for this study because of its ability to effectively capture complex dependencies among features through a probabilistic framework. This makes it particularly suitable for an IDS, as it enhances the system's capability to interpret the context of incoming traffic patterns by accurately modeling the correlations and relationships between different features. A notable strength of CRF is their capacity to rank features based on the sequential nature of traffic data. This ranking process is vital for the early detection of anomalies,

Algorithm 1 | Feature Engineering Algorithm (FEA)

```
Input: Dataset D with features $F = \{f_1, f_2, \ldots ,f_n\}$ and class labels C.

Output: Ranked feature set $F_{ranked}$.

Phase 1: Normalization

1. For each feature $f_i \in F$:
   a. If $f_i$ is a symbolic feature then map to numeric values.
   b. Log transformation was applied using Equation (1):
        $f_i \leftarrow \log(f_i + 1)$                                   (1)
   c. Apply min-max normalization using Equation (2):
        $f_i' = (b - a)\frac{f_i - \min(f_i)}{\max(f_i) - \min(f_i)} + a$     (2)
      where $[a, b]$ is the normalization range.
2. Return the transformed features:
        $F_{\text{transformed}} = \{f_1', f_2', \ldots, f_n'\}$

Phase 2: CRF based Ranking Method

1. Calculate Information Gain (IG) for each feature using Equation (3):
        $IG(f_i \mid C) = H(C) - H(C \mid f_i)$                          (3)
2. Filter the features satisfying $IG\ (f_i) \geq IG_{\text{threshold}}$ as $F_{\text{filtered}} \subseteq F_{\text{transformed}}$
3. Train the CRF model on the filtered and normalized feature set ($F_{\text{filtered}}$, C).
   The conditional probability is given by Equation (4):
        $P_{\theta(Y|X)} = \frac{1}{Z_{\theta(X)}} * \exp\left[\sum_{t=1}^{T}\sum_{k}\theta_k f_{k(y_{t-1}, y_t, X, t)}\right]$     (4)
   where $Z_\theta(X)$ is the partition function ensures normalization, and $\theta = \{\theta_k\}$ are model parameters estimated by maximizing the normalized log-likelihood as shown in Equation (5):
        $L(\theta) = \sum_{i=0}^{N} \log P_\theta\left(Y^{(i)} \mid X^{(i)}\right) - \lambda \mid \theta \mid_2^2$     (5)
   where λ is the regularization coefficient.
4. Aggregate CRF weights per feature.
   For each feature $f_j$, identify the set of CRF feature-function indices $K_j$ that depend on $f_j$. The importance of feature $f_j$ is given by Equation (6):
        $S_j = \sum_{k \in K_j} \mid \theta_k \mid$                          (6)
   where each captures the contribution of $f_j$ in the sequence-labeling process.
5. The CRF scores are normalized, as shown in Equation (7), to compute the combined ranking:
        $Score(f_j) = \frac{S_j}{\sum_{j'=1}^{n} S_{j'}}$                      (7)
6. The CRF score was combined with IG. The final combined rank is given as Equation (8):
        $\text{Rank}(f_j) = \alpha \bullet \text{norm}(\text{IG}_j) + (1 - \alpha) \bullet \text{norm}(S_j)$     (8)
   where norm (·) represents min–max normalization applied across all feature scores to scale values into the range [0,1]. The weighting parameter α ∈ [0,1] balances statistical relevance (IG) and contextual dependence (CRF); it is set to 0.5 by default.
7. Sort the features in descending order of Rank ($f_j$).
8. Select the top-K or those with Rank ($f_j$) exceeding a predefined threshold.
9. Return the ranked feature set $F_{ranked}$.
```

significantly increasing the sensitivity of the IDS to unusual patterns of behavior found within network traffic. In this study, Algorithm 1 outlines the implementation of both the normalization process and CRF-based feature selection strategy. In phase 1, the algorithm generates a transformed vector of feature values, reflecting the processed data. At the end of phase 2, it provides a ranked list of features prioritized according to their relevance in detecting anomalies. The ranked features produced by the CRF serve as essential inputs for the subsequent classification stage. This classification process is crucial for accurately identifying and categorizing detected intrusions. It is important to note that throughout this study, the dataset containing the pertinent features will be meticulously curated and utilized as input for further analysis, ensuring that the IDS is well-equipped to function optimally in real-world scenarios.

Classification

This subsection explains the newly proposed classifier for detecting intruders by using ML and DL algorithms such as Fuzzy SVM and CNN with temporal features. The analysis of temporal features encompasses changes that occur within traffic patterns. The initial step requires data preprocessing to extract temporal features through time-series traffic flow sequences using time windows which generate feature vectors for each input sample.

The proposed IDS uses temporal feature construction to capture sequential dependencies and short-term behavioral differences in network traffic. Network flows were divided into fixed-length windows of 5-second as a trade-off between computing overhead and temporal resolution. A stride of 2 seconds is used to ensure that two consecutive segments partially overlap, thereby maintaining continuity and preserving short-term attack dynamics.

The packet-level statistics such as flow duration, byte counts, and inter-arrival time are used to compute statistical aggregates such as the mean, variance, entropy, and maximum. For each window, a single temporal feature vector is formed by concatenating the aggregated descriptors with the static flow-level attributes.

$$F_t = \mathrm{Agg}\left(\{f_{1t}, f_{2t}, \ldots, f_{mt}\}\right)$$

where Agg () is the aggregation operator used in time window t. To remove label leakage, the annotation of each window is performed independently, based on the attack or benign status of the traffic seen in this window, independently of any future packets. Experimental analysis of window and stride parameters showed that short windows less than 3 seconds made the transient noise more sensitive and long windows beyond 10 seconds blurred transient bursts of attacks. The selected parameters of the 5-second window and 2-second stride gave the best trade-off between detection and response latency.

The model preserves the natural sequence of events, which helps identify anomalies by retaining essential information. The CRF method tracks temporal relationships among features in addition to its role in feature relevance scoring. This helps in ranking the most critical features based on their contextual and time-based relevance. The ranked sequences are fed into a CNN for training, which learns temporal traffic dynamics. The integrated components enable the system to assess evolving traffic patterns for precise detection of intrusions that occur throughout different attack situations.

FSVM helps to reduce the noise and uncertainty in feature selection with the aim of improving the accuracy of the classification. In this regard, we first used k-means clustering on the reduced feature vectors after CRF-based ranking, and calculated the fuzzy membership value of each sample, using its distance to the assigned clustering centroid. For each sample x_i, the fuzzy membership $u_i \in (0,1]$ is calculated based on its distance from its assigned centroid c_j using Equation (9).

$$u_i = \exp\left(-\frac{\left(d\left(x_i, c_j\right)\right)^2}{2\sigma_j^{\,2}}\right) \tag{9}$$

where $d(x_i, c_j)$ denotes the Euclidean distance between sample x_{i} and its centroid c_j. The parameter σ_j represents the dispersion (or spread) of cluster C_j and is calculated as the mean intra-cluster distance as shown in Equation (10):

$$\sigma_j = \frac{1}{\left|C_j\right|}\sum_{x \in C_j} \| x - c_j \| \tag{10}$$

These memberships serve as reliability weights, and are integrated into the FSVM optimization problem as weighted slack penalties. This ensures that unclear or noisy samples do not have a significant influence on the separating hyperplane. The decision of FSVM optimization function can be stated as shown in Equation (11):

$$\min_{w,b,\xi} \frac{1}{2} \| w \|^2 + C\sum_{i=1}^{n} u_i \xi_i \tag{11}$$

subject to constraints: $y_i(w^{\mathrm{T}}\phi(x_i) + b) \geq 1 - \xi_i$," " $\xi_i \geq 0$. where $u_i \in (0,1]$ represents the fuzzy membership value derived from k-means clustering. C is the penalty parameter that controls the trade-off between the classification error and margin maximization and $\phi(\cdot)$ denotes the nonlinear mapping function to the higher-dimensional feature space.

The FSVM results were further optimized using a decision tree trained jointly on the FSVM decision scores and selected features. Each stage complements one another. The k-means clustering helps to identify dependable data samples. The FSVM utilizes the fuzzy memberships to achieve robust classification. The decision tree incorporates interpretable rules that further enhance the accuracy and transparency. Algorithm 2 summarizes the entire integration procedure of the k-means clustering, fuzzy membership estimation, FSVM maximization, and refinement of decicion trees.

The combination of k-means, fuzzy rules, and a decision tree with FSVM was selected to address the uncertainty and nonlinearity in the network intrusion records. SVM treats all samples equally and, therefore, is prone to overfitting in the presence of outliers. Since fuzzy membership values are assigned based on cluster distances, the FSVM gives emphasis on reliable samples while minimizing the impact of noisy data. The decision tree also makes decisions more interpretable by transforming numerical decision scores into clear rule-based explanations. This hybrid setup has a balanced trade-off between robustness, accuracy, and explainability, satisfying the main requirements of a real-time, and transparent intrusion-detection system.

After obtaining the decision rules and weighted outputs from FSVM, the process continues with the CNN phase for temporal feature learning and final classification.

This study employs a two-phase approach, as shown in Algorithm 3, to categorize record sets based on the values of the contributing features, which are identified using the feature engineering algorithm. In the first phase, the Fuzzy SVM is applied to categorize the initial level prediction by considering the differences between the record sets using Euclidean distance measurement formula in the process of k-means clustering and the decision made by incorporating the standard decision tree. The standard CNN algorithm was applied in phase two with the integration of fuzzy logic methods and time limits. The proposed IDS implements fuzzy logic and time constraints to determine final decisions on input record sets. The two-phase identification is used to detect both low-frequency (minority or covert) and high-frequency attacks. The technique mainly detects high-priority categories such as DoS/DDoS and Brute Force in the initial step, and the second stage focuses on low-frequency attacks such as worms, shellcode, infiltration, and web attacks. The IDS functions effectively through this classification system to detect different types of attacks.

Algorithm 2 | FSVM Integration Process

```
Input: Feature matrix $X$; class labels $y$; number of clusters $K$
Output: Weighted decision scores and interpretable decision rules
1. Apply CRF-based ranking to obtain the reduced feature set $X_{reduced}$.
2. Perform k-means clustering on $X_{reduced}$ to obtain cluster centroids $C = \{c_1, c_2, \ldots, c_K\}$.
3. For each sample $x_i$, compute fuzzy membership $u_i$ using Equation (9) and cluster dispersion $\sigma_j$ using Equation (10).
4. The FSVM is trained using the weighted objective defined in Equation (11).
5. The FSVM decision scores for all samples were obtained.
6. A Decision Tree classifier was trained using the input features [$X_{\text{reduced}}$, FSVM$_{\text{scores}}$] to derive interpretable decision rules.
7. The final class labels and interpretable decision rules are output.
```

A dual-layered intrusion detection algorithm is executed according to the following steps:

Algorithm 3 | Fuzzy SVM and CNN with Temporal Logic Classifier (FS-CNN)

```
Input: Dataset D with selected features $F_{ranked}$
Output: Predicted attack categories $Y$
Phase 1: Fuzzy SVM with K-Means Clustering
    1. Read input data D($F_{ranked}$, C).
    2. Apply K-Means clustering algorithm for grouping the samples by applying Euclidean distance.
    3. Fuzzy rules are applied to assign samples to the initial clusters.
    4. Apply the decision tree for predicting the records as "Normal" and "Attacks."
    5. Partition the record sets into their respective groups-Normal, DoS/DDoS and Other Attacks.
Phase 2: Fuzzy Temporal Logic incorporated CNN.
    1. For each partition, a temporal feature sequence is generated.
    2. Perform the convolutional operation on the input sequences.
    3. Perform feature mapping.
    4. Perform the pooling operation on the feature maps.
    5. Determine the stride values and perform the corresponding operations.
    6. Apply ReLU activation followed by the SoftMax layer on the temporal features.
    7. Classify the samples in the fully connected layer into attack families defined in the CICIDS2017 and UNSW-NB15.
    8. Store and return the output predicted attack categories $Y$.
```

Results and Discussion

Dataset

The proposed IDS uses two benchmark datasets such as CICIDS2017 and UNSW-NB15 for evaluating the performance.

CICIDS2017 Dataset

The CICIDS2017 dataset[25] was created by Canadian Institute for Cybersecurity IDS, and it contains up-to-date network attacks and live attacks. The dataset contained 2,830,743 instances with 79 features. Moreover, it contains various attacks such as DOS, DDOS, Web-based, Brute force, Infiltration, Scan, Bot, and Heartbleed.

UNSW-NB15 Dataset

The UNSW-NB15 dataset was developed in the Cyber Range Lab at the University of New South Wales (UNSW) Canberra using the IXIA PerfectStorm tool to generate a blend of modern, realistic benign network traffic and synthetic contemporary attack activities. It has more than 2.5 million network records with approximately 175,341 and 82,332 records in the training and testing splits respectively. The records have 49 engineered features that include the flow, basic, content, and time-based attributes and a label column that identifies the traffic as either normal (0) or attack (1). In addition, attack_cat field identifies the attack family that corresponds to it; this has nine categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

Data Splitting and Validation Protocol

All datasets were divided into 70% training, 15% validation and 15% independent test sets, stratified by the distribution of classes. In CICIDS2017, temporal leakage was prevented by not allowing traffic captures on the same day to be included in the training and test sets simultaneously. In the case of UNSW-NB15, source file divisions were honored to avoid overlapping. To further handle the class imbalance in both datasets, the Synthetic Minority Oversampling Technique (SMOTE) and class-weighted loss functions were applied to the training set.

Within the training set, we performed 5-fold cross-validation repeated 3 times with different random seeds (15 runs) to tune hyperparameters and evaluate the model stability. The reported results were averaged over these runs. For final evaluation, each model was trained five times with different weight initializations, and the mean along with their standard deviation of the metrics were reported to account for stochastic variability.

Evaluation Parameters

The standard evaluation parameters such as precision, recall, F1-score and detection accuracy are considered in this work for evaluating the proposed IDS. The precision, recall and F1-score were calculated using the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values. The precision, recall, F1-score and Detection Accuracy (DA) are calculated using Equations 12–14.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

$$\text{F1} - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

$$\text{DA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{15}$$

The following hyperparameters were used in the proposed FSVM–CNN model:

CRF: L1 regularization = 0.1, L2 regularization = 0.1, max iterations = 100.

FSVM: Kernel = RBF, penalty parameter (C) = 10, gamma = 0.01. Membership function was Gaussian, derived from the distance of samples to their respective cluster centers.

CNN: Input consisted of the reduced feature vector from the CRF feature ranking. The architecture consisted of Conv1D (64 filters, kernel size = 3), followed by MaxPooling, Conv1D (128 filters, kernel size = 3), Dense (128 units, ReLU activation), and a Softmax output layer. We trained with Adam optimizer (learning rate = 0.001, batch_size = 128, epochs = 50).

All the baseline algorithms were trained and tested under the same preprocessing and parameterization conditions. The continuous attributes were z-score normalized and the categorical attributes were uniformly label-encoded across data sets. Hyperparameters were optimized using 5-fold cross-validation to minimize overfitting and bias when comparing each baseline. In the case of SVM, the penalty term is, in particular, C = 1.0 and RBF kernel with γ = 0.01. The values were γ = 0.01; with Random Forest, estimators = 200 and depth = 12; for CNN and LSTM models, learning rate = 10^{-3}, batch size = 64, and Adam optimizer with early stopping was used. All models shared the same train-test splits (70%–30%) and random seeds to ensure reproducibility.

Although recent graph-based intrusion detection frameworks such as GCN and GraphSAGE have shown promise in modeling relational network dependencies, they were not experimentally included in this study because of their high computational cost and data-graph transformation requirements. These approaches are reserved for future exploration to complement the proposed FSVM–CNN model.

Experimental Results

The performance of the IDS was evaluated using two benchmark datasets, CICIDS2017 and UNSW-NB15, on a system with Intel Core i5 processor, 16 GB RAM and 1TB storage. The performance was assessed in terms of detection accuracy, precision, recall and F1-score using both reduced and full-featured datasets. Experiments were conducted to systematically evaluate: (i) baseline performance of classical ML and standard DL models, (ii) the effect of applying the proposed IG-R+CRF feature selection, (iii) the performance of recent advanced models such as GAN-IDS, Transformer-based IDS, and Federated IDS, and (iv) the superiority of the proposed hybrid FSVM–CNN.

Tables 1 and 2 present the detection accuracies of various classifiers across individual attack categories on the CICIDS2017 and UNSW-NB15 datasets.

Compared to SVM and RF, the detection accuracies of CNN and LSTM is significantly higher. The GAN-based IDS provides additional improvements, which

Table 1 | Detection accuracy per attack type on CICIDS2017 dataset

Attack Type	SVM	RF	CNN	LSTM	GAN-IDS	Transformer IDS	Federated IDS	Proposed FSVM-CNN
DoS/DDoS	94.2	95.1	96.0	96.5	96.8	97.3	97.7	99.1
Brute Force	93.8	94.9	95.7	96.2	96.6	97.1	97.5	99.0
Infiltration	93.5	94.6	95.4	96.0	96.5	96.8	97.4	99.2
Web Attacks	93.9	94.7	95.6	96.1	96.6	97.0	97.6	99.1
Botnet/Mirai	93.6	94.5	95.5	96.0	96.5	96.9	97.5	99.2
Normal	94.5	95.2	96.2	96.7	97.0	97.5	98.0	99.3

Table 2 | Detection accuracy per attack type on UNSW-NB15 dataset

Attack Type	SVM	RF	CNN	LSTM	GAN-IDS	Transformer IDS	Federated IDS	Proposed FSVM-CNN
Fuzzers	88.5	90.2	91.0	91.7	92.4	93.0	93.8	97.8
Analysis	89.0	90.5	91.4	92.0	92.7	93.3	94.0	98.0
Backdoors	88.8	90.4	91.3	91.9	92.5	93.1	93.9	97.9
DoS	89.2	90.8	91.6	92.2	92.8	93.4	94.1	98.1
Exploits	88.6	90.3	91.1	91.8	92.5	93.2	93.8	97.9
Generic	89.5	91.0	91.8	92.4	93.1	93.6	94.2	98.2
Reconnaissance	89.3	90.9	91.7	92.3	92.9	93.5	94.1	98.1
Shellcode	88.7	90.2	91.0	91.7	92.4	93.0	93.7	97.8
Worms	89.1	90.7	91.5	92.1	92.8	93.3	93.9	98.0
Normal	90.0	91.5	92.2	92.9	93.5	94.0	94.6	98.4

Table 3 | Performance metrics on CICIDS2017 dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Detection Rate (%)	False Alarm Rate (%)	AUC
SVM	97.0	96.8	96.5	96.6	96.5	3.2	0.982
Random Forest	97.4	97.2	97.0	97.1	97.0	2.8	0.985
CNN	97.6	97.5	97.3	97.4	97.3	2.5	0.988
LSTM	97.9	97.8	97.5	97.6	97.5	2.2	0.989
GAN-IDS	98.0	97.9	97.7	97.8	97.7	2.1	0.991
Transformer IDS	98.2	98.1	97.9	98.0	97.9	2.0	0.993
Federated IDS	98.4	98.2	98.1	98.1	98.1	1.7	0.995
Proposed FSVM-CNN	99.0	98.8	98.7	98.7	98.7	1.5	0.996

Table 4 | Performance metrics on UNSW-NB15 dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Detection Rate (%)	False Alarm Rate (%)	AUC
SVM	96.0	95.8	95.6	95.7	95.6	3.5	0.975
Random Forest	96.6	96.5	96.2	96.3	96.2	3.0	0.979
CNN	96.9	96.8	96.5	96.6	96.5	2.7	0.981
LSTM	97.3	97.2	96.9	97.0	96.9	2.4	0.985
GAN-IDS	97.5	97.4	97.2	97.3	97.2	2.3	0.986
Transformer IDS	97.8	97.7	97.5	97.6	97.5	2.1	0.988
Federated IDS	98.0	97.9	97.8	97.8	97.8	1.9	0.991
Proposed FSVM-CNN	98.6	98.5	98.4	98.3	98.4	1.6	0.993

are further enhanced by Transformer- and Federated-IDS approaches. However, the proposed FSVM-CNN outperforms all the baseline models and achieves more than 99% accuracy on CICIDS2017 and over 97% on UNSW-NB15, demonstrating its stability and high generalization to different classes of attacks.

Tables 3 and 4 present the overall performance metrics of different IDS models. The results show that GAN-IDS performs better than CNN and LSTM, whereas transformer IDS demonstrates superior detection, and federated IDS provides extra benefits to accuracy and AUC with distributed learning.

The macro-average AUC was calculated as the arithmetic mean of the respective class-wise AUC values of each attack category as shown in Equation (16). This measure represents a balanced view of the model's ability to discriminate between all types of attack.

$$\mathrm{AUC}_{\mathrm{macro}} = \left(\frac{1}{K}\right)\Sigma_{i=1}^{K}\mathrm{AUC}_{i} \tag{16}$$

The False Positive Rate (FPR) for each class was calculated as shown in Equation (17):

$$\mathrm{FPR}_{i} = \frac{(\mathrm{FP}_{i})}{\mathrm{FP}_{i} + \mathrm{TN}_{i}} \tag{17}$$

and were then averaged to obtain the overall FPR. This formulation also ensures that all attack categories are equally represented by the aggregate measure, eliminating the dominance of majority classes.

The proposed FSVM-CNN achieved the best results across all metrics and datasets, with the highest accuracy, precision, recall, F1-score, and AUC, and the lowest false alarm rate, confirming its superiority over state-of-the-art IDS approaches. The performance of the FSVM-CNN in terms of evaluation metrics is highlighted in Tables 3–5 further shows the results of statistical significance testing, demonstrating that the proposed model outperforms existing models.

Table 5 | Statistical significance analysis of proposed FSVM-CNN IDS compared with baselines

Baseline Model	P-value	95% CI (Accuracy Difference)
FSVM-CNN vs. SVM	<0.001	[+1.5%, +2.1%]
FSVM-CNN vs. RF	<0.001	[+1.2%, +1.9%]
FSVM-CNN vs. CNN	<0.001	[+0.8%, +1.5%]
FSVM-CNN vs. LSTM	0.004	[+0.6%, +1.2%]
FSVM-CNN vs. GAN-IDS	0.011	[+0.4%, +0.9%]
FSVM-CNN vs. Transformer IDS	0.021	[+0.3%, +0.8%]
FSVM-CNN vs. Federated IDS	0.037	[+0.2%, +0.6%]

The main limitations of earlier hybrid IDSs[6,10,16] are as follows: relies solely on filter-based selection without modeling dependencies,[6] applies PSO to search architecture but does not consider feature uncertainty,[10] and couples CNN with metaheuristics but does not refine fuzzy boundaries.[16] In contrast, the proposed FSVM-CNN model integrates IGR and CRF to rank features, fuzzy membership to manage uncertainty, and CNN to learn temporal-spatial features. As shown in Table 6, the ablation study measures the contribution of each component. Table 6 shows that the integration of IGR+CRF with the FSVM–CNN model achieved the highest accuracy and AUC, highlighting the

Table 6 | Ablation study of FSVM–CNN components on CICIDS2017 and UNSW-NB15 datasets

Configuration	CICIDS2017		UNSW-NB15	
	Accuracy (%)	AUC	Accuracy (%)	AUC
CNN only	97.6	0.988	96.9	0.981
FSVM only	97.8	0.989	97.1	0.983
IGR-only + FSVM-CNN	98.4	0.993	97.8	0.988
CRF-only + FSVM-CNN	98.7	0.995	98.2	0.991
Proposed IGR+CRF + FSVM–CNN (full)	99.0	0.996	98.6	0.993

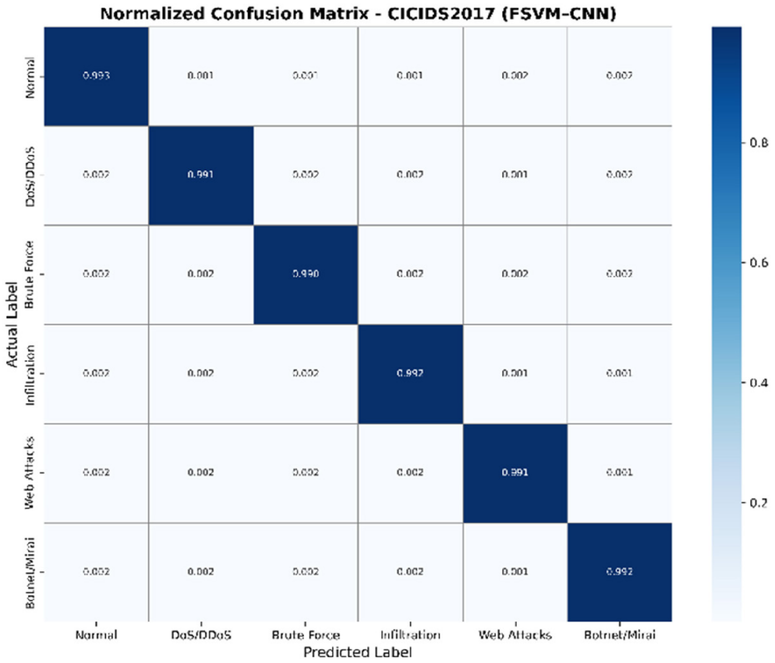

Fig 2 | Normalized confusion matrix of FSVM-CNN on CICIDS2017

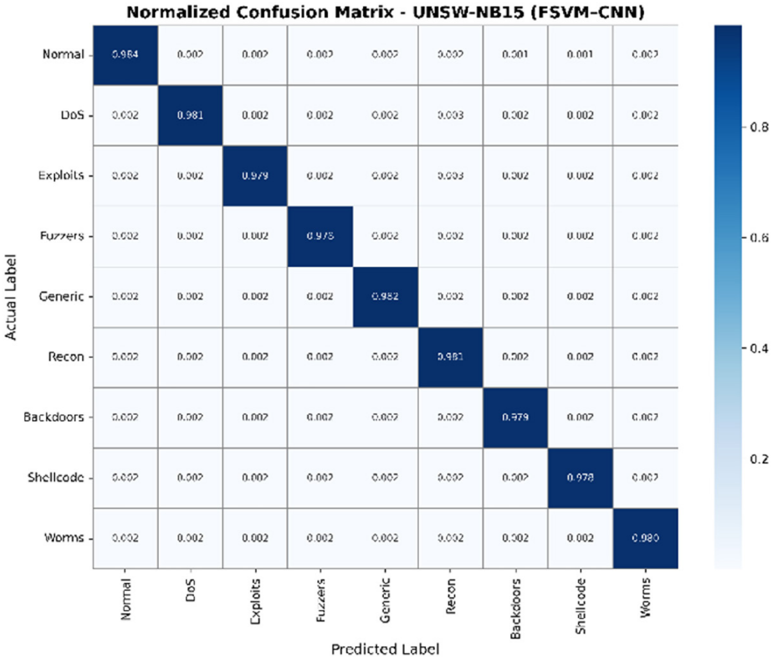

Fig 3 | Normalized confusion matrix of FSVM-CNN on UNSW-NB15

effectiveness of combining IGR and CRF for feature enhancement, resulting in superior intrusion detection.

To further analyze the class-wise detection behavior, the per-class Precision, Recall, and F1-score were computed for each type of attack in each dataset. The mean values and 95% confidence intervals (CIs) were estimated using bootstrap resampling over five independent experimental runs. The experimental results indicate that the proposed FSVM-CNN model provides consistently stable results. The performance variation was within ±0.5% for most attack classes, demonstrating the reliability of the model's prediction. For example, in the CICIDS2017 data, the DoS and Infiltration classes had F1-scores of 99.1 + 0.3 and 98.4 + 0.6, respectively, whereas in UNSW-NB15, the Generic and Exploits classes had F1-scores of 97.8 + 0.5 and 95.6 + 0.4, respectively. These repeatable findings prove the strength of the proposed IDS, which is resistant to different types of attacks.

The most discriminative and non-redundant attributes of each dataset were identified in the proposed model. Based on 78 initial features in CICIDS2017 and 45 features in UNSW-NB15, the top-ranked subsets retained 32 and 27 features, respectively. The representative attributes retained from both datasets are as follows: Flow_Duration, Flow_Bytes/s, Packet_Length_Mean, and Subflow_Fwd_Packets for CICIDS2017, and dur, sbytes, sttl, and ct_state_ttl for UNSW-NB15.

In addition to the accuracy-based analysis, the runtime performance of the proposed FSVM-CNN model was evaluated to ensure its suitability for real-time intrusion detection. All experiments were conducted on a Core i5 CPU with 16 GB RAM, identical to the experimental setup used for accuracy assessment. The average training times of the proposed model on CICIDS2017 and UNSW-NB15 were 342 and 297 seconds, respectively. The framework was able to process approximately 9,800 samples per second with an average inference latency of 0.63 ms per sample, demonstrating its capability to handle moderate-to-heavy traffic in real time.

The normalized confusion matrices for CICIDS2017 and UNSW-NB15 are shown in Figures 2 and 3, respectively. The results highlight the per-class classification capability of the proposed model, achieving accuracies above 0.990 for all classes in the CICIDS2017 dataset and ranging between 0.978 and 0.984 for all classes in the UNSW-NB15 dataset.

Overall, the evaluation on CICIDS2017 and UNSW-NB15 consistently shows that the proposed FSVM–CNN outperforms state-of-the-art models in terms of detection accuracy, precision, recall, F1-score, and AUC while maintaining a lower false alarm rate. The ablation analysis emphasizes the significance of combining IGR and CRF for feature selection and the confusion matrices confirm reliable per-class recognition across a various attack categories. Collectively, these results confirm the robustness and generalizability of the proposed model. Furthermore, the FSVM–CNN demonstrated good training and inference performance on standard hardware, with IGR and CRF reducing dimensionality and overhead, confirming its suitability for real-time IDS deployment on GPU-enabled or embedded platforms.

Conclusion and Future Work

This study introduces a novel feature engineering approach and a combination of CNN and fuzzy SVM classifiers with temporal characteristics to create a new IDS that efficiently predicts and detects intrusions. The proposed model, which uses a combination of IGR and CRF to rank features, fuzzy membership functions to deal with uncertainty, and CNN to learn temporal-spatial patterns, resulted in state-of-the-art performance across the CICIDS2017 and UNSW-NB15 datasets. The system achieved a high accuracy of more than 99% on CICIDS2017 and more than 98% on UNSW-NB15, with better precision, recall, and AUC and a lower false alarm rate than the baseline models. Its superiority was further confirmed by statistical significance testing. The per-class confusion matrices further validated that the proposed IDS can reliably distinguish between diverse attack categories such as DoS/DDoS, brute force, infiltration, web attacks, botnet (CICIDS2017), fuzzers, analysis, backdoors, exploits, reconnaissance, shellcode, worms, and generic attacks (UNSW-NB15). This highlights the model's ability to generalize across both frequent and diverse patterns of attack.

Despite these strengths, this study has certain limitations. Although the false alarm rate is lower than that of baseline models, even a small number of false positives can have significant consequences. Moreover, experimental evaluations were performed out on benchmark datasets, and real-world deployment may involve greater complexities. Furthermore, ensuring data privacy and preserving operational trust are essential for reliable real-world deployment.

Future work can focus on optimizing the detection accuracy and efficiency by utilizing advanced feature selection techniques such as Particle Swarm Optimization (PSO) or Genetic Algorithm (GA). In addition, federated learning can be incorporated to develop privacy-preserving and collaborative IDS models, while explainable AI (XAI) can be employed to further enhance interpretability and trust in real-world applications.

Supplementary Material

The supplementary materials are available online.

References

1 El-Khatib K. Impact of feature reduction on the efficiency of wireless intrusion detectIon systems. IEEE Trans Parallel Distrib Syst. 2010;21(8):1143–9. https://doi.org/10.1109/TPDS.2009.142

2 Mabu S, Chen C, Lu N, Shimada K, Hirasawa K. An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. IEEE Trans Syst Man Cybern C Appl Rev. 2011;41(1):130–9. https://doi.org/10.1109/TSMCC.2010.2050685

3 Hu W, Hu W, Maybank S. AdaBoost-based algorithm for network intrusion detection. IEEE Trans Syst Man Cybern B Cybern. 2008;38(2):577–83. https://doi.org/10.1109/TSMCB.2007.914695

4 Vinayakumar R, Soman KP, Poornachandran P. Applying convolutional neural network for network intrusion detection. Proc Int Conf Adv Comput Commun Inform. 2017;1222–8. https://doi.org/10.1109/ICACCI.2017.8126009

5 Sadioura JS, Singh S, Das A. Selection of sub-optimal feature set of network data to implement machine learning models to develop an efficient NIDS. Proc Int Conf Data Sci Eng. 2019;120–5. https://doi.org/10.1109/ICDSE47409.2019.8971479

6 Kasongo SM, Sun Y. A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access. 2019;7:38597–607. https://doi.org/10.1109/ACCESS.2019.2905633

7 Mendonça RV, Teodoro AAM, Rosa RL, Saadi M, Melgarejo DC, Nardelli PHJ, et al. Intrusion detection system based on fast hierarchical deep convolutional neural network. IEEE Access. 2019;9:61024–34. https://doi.org/10.1109/ACCESS.2021.3074664

8 Dutt I, Borah S, Maitra IK. Immune system based intrusion detection system (IS-IDS): A proposed model. IEEE Access. 2020;8:34929–41. https://doi.org/10.1109/ACCESS.2020.2973608

9 Lopez-Martin M, Carro B, Sanchez-Esguevillas A. Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Syst Appl. 2020;141:112963. https://doi.org/10.1016/j.eswa.2019.112963

10 Elmasry W, Akbulut A, Zaima AH. Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Comput Netw. 2020;168:107042. https://doi.org/10.1016/j.comnet.2019.107042

11 Kasongo SM, Sun Y. A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Comput Secur. 2020;92:101752. https://doi.org/10.1016/j.cose.2020.101752

12 Azizjon M, Jumabek A, Kim W. 1D CNN based network intrusion detection with normalization on imbalanced data. Proc Int Conf Artif Intell Inf Commun. 2020;218–24. https://doi.org/10.1109/ICAIIC48513.2020.9064976

13 Manikandan V, Gowsic K, Prince T, Umamaheswari R, Ibrahim BF, Sampathkumar A. DRCNN-IDS approach for intelligent intrusion detection system. Proc Int Conf Comput Inf Technol. 2020;1–4. https://doi.org/10.1109/ICCIT-144147971.2020.9213779

14 Wisanwanichthan T, Thammawichai M. A double-layered hybrid approach for network intrusion detection system using combined naive Bayes and SVM. IEEE Access. 2021;9:138432–50. https://doi.org/10.1109/ACCESS.2021.3118573

15 Kunang YN, Nurmaini S, Stiawan D, Suprapto BY. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. J Inf Secur Appl. 2021;58:102804. https://doi.org/10.1016/j.jisa.2021.102804

16 Fatani A, Abd Elaziz M, Dahou A, Al-Qaness MAA, Lu S. IoT intrusion detection system using deep learning and enhanced transient search optimization. IEEE Access. 2021;9:123448–64. https://doi.org/10.1109/ACCESS.2021.3109081

17 Ho S, Jufout SA, Dajani K, Mozumdar M. A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network. IEEE Open J Comput Soc. 2021;2:14–25. https://doi.org/10.1109/OJCS.2021.3050917

18 Li B, Wu Y, Song J, Lu R, Li T, Zhao L. DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Trans Ind Inform. 2021;17(8):5615–24. https://doi.org/10.1109/TII.2020.3023430

19 Lin K, Xu X, Xiao F. MFFusion: A multi-level features fusion model for malicious traffic detection based on deep learning. Comput Netw. 2022;202:108658. https://doi.org/10.1016/j.comnet.2021.108658

20 Zhang X, Yang F, Hu Y, Tian Z, Liu W, Li Y, et al. RANet: Network intrusion detection with group-gating convolutional neural network. J Netw Comput Appl. 2022;198:103266. https://doi.org/10.1016/j.jnca.2021.103266

21 Zhao X, Fok KW, Thing VLL. Enhancing network intrusion detection performance using generative adversarial networks. arXiv preprint. 2024;arXiv:2404.07464. https://doi.org/10.48550/arXiv.2404.07464

22 Xi C, Wang H, Wang X. A novel multi-scale network intrusion detection model with transformer. Sci Rep. 2024;14(1):23239. https://doi.org/10.1038/s41598-024-74214-w

23 Adjewa F, Esseghir M, Merghem-Boulahia L. Efficient federated intrusion detection in 5G ecosystem using optimized BERT-based model. arXiv preprint. 2024;arXiv:2409.19390. https://doi.org/10.48550/arXiv.2409.19390

24 Saheed AI, Abiodun S, Misra S, Holone MK, Colomo-Palacios R. A machine learning-based intrusion detection for detecting Internet of Things network attacks. Alexandria Eng J. 2022;61:9395–409. https://doi.org/10.1016/j.aej.2022.02.063

25 Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proc Int Conf Inf Syst Secur Privacy. Cham: Springer; 2018. p. 108–16. https://doi.org/10.5220/0006639801080116