

# Cyber Attack Detection and Classification Using Transfer Learning with Convolutional Neural Networks

Dr. Koduru Prasadarao<sup>1</sup>; Katikam Mahesh<sup>2</sup>

<sup>1</sup>Professor, Department of CSE, PACE Institute of Technology & Sciences (Autonomous), NH-5, Near Valluramma Temple, Ongole, Valluru, Andhra Pradesh 523272, India

<sup>2</sup>Assistant Professor, Department of CSE, BVRIT Institute of Technology, Narsapur, Vishnupur, JNTUH, India

<sup>2</sup>ORCID: <https://orcid.org/0009-0000-0707-111>

Publication Date: 2026/09/30

**Abstract:** Cyber attacks continue to evolve in volume, diversity, and complexity, making timely identification of malicious network activity an important cybersecurity requirement. This paper presents a deep learning framework for cyber attack detection and multi-class classification using transfer learning with convolutional neural networks. The proposed approach converts structured network-flow features into a representation suitable for convolutional feature extraction and adapts a pretrained convolutional backbone to the cybersecurity classification task. The workflow includes data preprocessing, feature normalization, class-label preparation, transfer learning, supervised fine-tuning, and evaluation using accuracy, precision, recall, F1-score, and confusion matrices. The approach is designed to reduce training requirements compared with learning a deep model entirely from random initialization while retaining the representation-learning capability of convolutional networks. An experimental protocol using a publicly available intrusion-detection dataset is described so that the results can be reproduced and compared with conventional machine-learning and non-transfer-learning deep models. The study also discusses class imbalance, domain shift, computational cost, and the need for careful validation before deployment in operational security environments.

**Keywords:** Cybersecurity; Intrusion Detection; Cyber Attack Classification; Transfer Learning; Convolutional Neural Network; Deep Learning; Network Traffic; Intrusion Detection System.

**How to Cite:** Dr. Koduru Prasadarao; Katikam Mahesh (2026) Cyber Attack Detection and Classification Using Transfer Learning with Convolutional Neural Networks. *International Journal of Innovative Science and Research Technology*, 11(9), 1876-1879. <https://doi.org/10.38124/ijisrt/26sep819>

## I. INTRODUCTION

Modern computer networks generate large volumes of heterogeneous traffic, and security monitoring systems must distinguish legitimate activity from malicious behavior. Traditional intrusion detection systems commonly rely on manually engineered features, statistical thresholds, signatures, or classical machine-learning algorithms. These methods can be effective for known patterns but may require substantial feature engineering and can face difficulties when attack behavior changes.

Deep learning provides an alternative by learning hierarchical representations from data. Convolutional neural networks (CNNs), although originally developed for visual data, can extract local patterns from suitably arranged feature representations. Transfer learning further allows a model trained on a source task to be adapted to a target task. In

cybersecurity, this idea can reduce the amount of task-specific training required and can provide a structured initialization for deep feature extraction.

This work investigates a transfer-learning CNN framework for cyber-attack detection and classification. The study focuses on a reproducible pipeline: preprocessing network-flow records, mapping the features to a CNN-compatible representation, adapting a pretrained backbone, fine-tuning selected layers, and evaluating multi-class predictions. The paper does not claim that transfer learning is universally superior; instead, it defines an experimental methodology for measuring its effectiveness under the chosen dataset and evaluation protocol.

## II. RELATED WORK

Research on intrusion detection has progressed from signature-based systems to statistical learning, ensemble methods, recurrent networks, autoencoders, and convolutional architectures. Classical algorithms such as decision trees, random forests, support vector machines, and k-nearest neighbors remain useful baselines because they can perform well on tabular network-flow data with limited computational cost.

Deep learning approaches reduce reliance on manually selected rules by learning representations directly from training data. CNNs can capture local correlations after features are arranged into structured tensors. Transfer learning has been extensively studied in domains where a pretrained representation can be adapted to a related target task. For cybersecurity, however, source and target domains can differ considerably, so the usefulness of transferred features should be verified experimentally rather than assumed.

Recent intrusion-detection studies also emphasize class imbalance, dataset age, duplicated records, changing traffic distributions, and the difference between laboratory benchmarks and operational traffic. Consequently, evaluation should report per-class metrics and not rely on accuracy alone.

## III. PROPOSED METHODOLOGY

### ➤ System Architecture

The proposed architecture contains five major stages: dataset acquisition, preprocessing, CNN-compatible representation generation, transfer learning and fine-tuning, and attack classification. A schematic flow is: Network-flow dataset → data cleaning → categorical encoding → feature scaling → class balancing → CNN input transformation → pretrained CNN backbone → task-specific dense layers → softmax output → predicted attack class.

The classification layer produces a probability distribution over the selected traffic classes. The class with the highest posterior probability is taken as the predicted label. For binary detection, the same framework can use a two-class output representing benign and attack traffic.

### ➤ Data Preprocessing

Records containing invalid or missing values are handled according to a predefined preprocessing policy. Categorical variables are encoded numerically, while continuous variables are scaled using statistics calculated from the training partition only. The target label is separated before transformation to prevent label leakage.

Because intrusion datasets frequently contain uneven class distributions, the training set may require class-weighted loss, controlled resampling, or another documented balancing strategy. Validation and test partitions should retain their natural class distribution whenever possible so that reported metrics reflect the evaluation population.

### ➤ CNN Representation

Network-flow data are normally tabular rather than image-based. To apply a CNN, the normalized feature vector can be mapped into a fixed two-dimensional feature grid or tensor. The transformation must be deterministic and identical during training and inference. The purpose is not to claim that network records are images, but to provide a structured input on which convolutional filters can learn local feature interactions.

### ➤ Transfer Learning

A pretrained CNN is used as the initial feature extractor. Early layers may be frozen during the first training stage, while the final classification head is replaced with layers corresponding to the attack classes. After stabilization of the new classifier, selected deeper layers can be unfrozen and fine-tuned using a smaller learning rate. This two-stage strategy can reduce destructive updates to general representations.

The transfer-learning pipeline can be expressed as: pretrained weights → remove source classifier → attach cybersecurity classifier → freeze selected layers → train classification head → unfreeze selected layers → fine-tune → evaluate on unseen test data.

### ➤ Classification and Loss Function

For a K-class problem, the final softmax layer produces K class probabilities. Cross-entropy loss is used for supervised classification. When class imbalance is significant, class weights can be incorporated into the loss so that errors on minority classes receive greater contribution. Hyperparameters such as learning rate, batch size, number of frozen layers, dropout, and number of fine-tuning epochs should be selected using the validation set and documented before final testing.

## IV. EXPERIMENTAL DESIGN

The recommended evaluation uses a publicly available intrusion-detection dataset containing benign and multiple attack categories. The dataset should be partitioned into training, validation, and independent test subsets without allowing duplicate or near-duplicate flows to cross partitions. A fixed random seed should be recorded for reproducibility.

Three model groups should be compared: (1) classical machine-learning baselines, (2) a CNN trained from random initialization, and (3) the proposed transfer-learning CNN. The comparison should use identical preprocessing and test data. This design isolates whether the benefit comes from the CNN architecture itself or from transfer initialization.

Table 1 Recommended Experimental Configuration

| Component      | Configuration                                     | Purpose                                 |
|----------------|---------------------------------------------------|-----------------------------------------|
| Input          | Normalized network-flow features                  | Stable model input                      |
| Representation | Fixed CNN-compatible tensor                       | Enable convolutional feature extraction |
| Backbone       | Pretrained CNN                                    | Transfer reusable representations       |
| Training       | Head training followed by fine-tuning             | Adapt model to attack classes           |
| Output         | Softmax multi-class classifier                    | Attack-category prediction              |
| Metrics        | Accuracy, Precision, Recall, F1, Confusion Matrix | Comprehensive evaluation                |

#### ➤ Evaluation Metrics

Accuracy is the proportion of correctly classified records among all evaluated records. Precision measures the proportion of predicted instances of a class that are correct, while recall measures the proportion of actual instances of a class that are detected. The F1-score is the harmonic mean of precision and recall. Because a high-frequency class can dominate aggregate accuracy, macro-averaged precision, recall, and F1-score should also be reported.

A confusion matrix provides class-by-class information about false positives and false negatives. For cybersecurity deployment, per-class recall is particularly informative

because missed malicious events can have operational consequences. However, the importance of each error type depends on the application and should be interpreted together with false-alarm rates and operational costs.

## V. RESULTS AND DISCUSSION

The experimental results should be reported only after executing the defined protocol on the selected dataset. The following table is intentionally provided as a results-reporting template rather than fabricated experimental evidence. This avoids presenting unmeasured performance as a research finding.

Table 2 Results Reporting Template

| Model                 | Accuracy       | Macro Precision | Macro Recall   | Macro F1       |
|-----------------------|----------------|-----------------|----------------|----------------|
| Classical baseline    | To be measured | To be measured  | To be measured | To be measured |
| CNN from scratch      | To be measured | To be measured  | To be measured | To be measured |
| Transfer-learning CNN | To be measured | To be measured  | To be measured | To be measured |

The principal analysis should compare convergence behavior, per-class performance, and generalization to unseen traffic. If transfer learning reduces the number of epochs or improves minority-class recognition, these effects should be reported with confidence intervals or repeated-run variation where feasible. Conversely, if the transferred representation performs worse than a model trained from scratch, possible explanations include domain mismatch between the source pretraining data and network-flow data, an unsuitable feature-to-grid mapping, or insufficient fine-tuning.

#### ➤ Security and Deployment Considerations

A practical intrusion-detection model must account for concept drift, encrypted traffic, adversarial manipulation, changing attack families, and false-positive costs. A model that performs well on a benchmark dataset may not retain the same performance on current operational traffic. Periodic evaluation on temporally separated data is therefore recommended.

Model outputs should also be integrated with security operations carefully. Automated blocking based solely on a probabilistic classifier can create availability risks when false positives occur. A layered architecture combining model predictions with policy controls, logging, analyst review, and other security signals can provide additional safeguards.

## VI. LIMITATIONS

The main limitation of the proposed methodology is that transfer learning depends on the relationship between the source representation and the target cybersecurity domain. CNN input transformation can also introduce artificial spatial relationships that are not inherent in tabular network data. Dataset bias, class imbalance, outdated attack patterns, and leakage between related records can further inflate measured performance. These limitations should be explicitly tested in future experiments.

## VII. CONCLUSION

This paper presented a transfer-learning framework using convolutional neural networks for cyber-attack detection and classification. The method combines controlled preprocessing, structured CNN input generation, pretrained feature extraction, task-specific classifier replacement, and selective fine-tuning. A reproducible evaluation protocol using per-class metrics and an independent test set was defined to compare transfer learning with conventional baselines and CNN training from random initialization. Future work can investigate temporal domain adaptation, lightweight models for edge deployment, explainable predictions, continual learning, and robustness against adversarially manipulated network traffic.

## REFERENCES

- [1]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [2]. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [3]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [4]. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
- [5]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. International Conference on Information Systems Security and Privacy*, pp. 108–116, 2018.
- [6]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.
- [7]. T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.
- [8]. N. Papernot et al., "The limitations of deep learning in adversarial settings," in *Proc. IEEE European Symposium on Security and Privacy Workshops*, pp. 372–387, 2016.