JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

End-to-End Deep Learning Framework for Intrusion Detection Using Raw Network Traffic Data

A. Kalaivani[1] and R. Pugazendi [2]

[1]Research scholar, PG & Research Department of Computer Science, Government Arts College (Autonomous), Salem-636007, Tamil Nadu, India,
[2]Assistant Professor, PG & Research Department of Computer Science, Government Arts College (Autonomous), Salem-636007, Tamil Nadu, India

Abstract

Intrusion Detection Systems plays an vital role for protecting the computer networks from the trending cyberattacks which is increasing drastically .The traditional machine learning based IDS is lacking and has many drawbacks in many ways. The machine learning based IDS completely depends heavily on the handcrafted features; it needs extensive preprocessing and requires explicit normalization to be done. This limits the ML models for IDS their ability to adapt to evolving attack patterns. The limitations this work focuses based on a deep learning-based intrusion detection system. This end to end deep learning based IDS operates directly on raw flow-level network traffic with minimal preprocessing. A Convolutional Neural Network (CNN) is employed to automatically learn the discriminative feature representations and perform attack classification. This paper proposes a model by selecting raw traffic selection included in the CSE-CIC-IDS2018 dataset and batch normalization layers are embedded within the CNN which handles the internal feature scaling and it also eliminates the need for the explicit data normalization. The proposed framework achieved an accuracy of 95.6% which has outperformed the traditional machine learning models. Thus this result establishes a strong baseline for IDS and it also motivates further using advanced feature learning and optimization strategies.

Keywords: Intrusion Detection System, Deep Learning, CNN, Automatic Feature Learning, CSE-CIC-IDS2018

1. Introduction

The widespread adoption of many trending technologies like cloud computing, Internet of things(IOT) and the high speed enterprise networks has drastically increased the exposure to cyber threats like brute-force-intrusion and denial of service attacks. Widely used intrusion detection systems (IDS) keep an overview on network traffic and detect malicious activity. Machine learning models that are conventional or the signature matching faces biggest challenges in detecting the novel and complex attack patterns [1], [4], [10]. The drawback presents in the Machine learning based IDS is it depends on the handcrafted features, manual processing and it requires explicit feature normalization. In addition, Deep Learning models, have shown an important ability to understand complicated and non-linear patterns straight from the raw data, opening the

way for their development. This paper aims to establish Deep learning as a foundational approach for IDS by proposing a very simple but effective end to end CNN based IDS which learns directly from minimally processed network traffic data.

Problem statement and research gap

Though the progress in IDS research is trending, several limitations remains in the traditional approaches. The manual feature engineering is the main drawback [7].It is difficult in modelling with high-dimensional and non-linear traffic patterns [1], [3], [4], [10]. The evolving new attacks are not identified. These research gap has paid way for the Deep learning based approach with raw traffic data ,minimal processing and also does not require explicit normalization.

2. Literature Review

Author (Year)	Aim of the Study	Key Contributions	Methodology & Datasets Used
Basati et al. (2022)	Develop a lightweight and accurate IDS suitable for resource-constrained IoT environments	Proposed a Parallel Deep Auto-Encoder (PDAE) architecture that reduces model parameters and memory usage while maintaining high detection accuracy	Parallel deep auto-encoders for feature representation followed by a classifier; evaluated on IoT-focused benchmark datasets
Caville et al. (2022)	Examine learning that is self to identify network intrusions	Introduced the first practical self-supervised GNN-based flow-level NIDS, improving generalization to unseen traffic without heavy reliance on labeled data	Graph construction from network flows → self-supervised GNN training → anomaly scoring; evaluated on benchmark NIDS datasets
Saurabh et al. (2022)	Model temporal patterns in IoT network traffic for intrusion detection	Designed LSTM-based and auto encoder-based IDS architectures achieving better accuracy and low-latency detection for IoT traffic	LSTM auto encoder and stacked/bidirectional LSTM classifiers; evaluated on UNSW-NB15 and BoT-IoT datasets
Altunay et al. (2023)	Improve detection of diverse attack types using hybrid deep learning models	Demonstrated that a CNN–LSTM hybrid architecture outperforms standalone models with reduced false alarm rates	CNN for the extraction of spatial features and LSTM for temporal modeling; experiments conducted on common IDS benchmark datasets
Debicha et al. (2023)	Detect adversarial and evasion attacks targeting DL-based IDS models	Proposed a transfer-learning-based multi-detector framework that enhances robustness against adversarial perturbed traffic	Transfer learning detectors trained on selected feature subsets; adversarial attack generation and detection evaluated on

			standard IDS datasets
Debicha et al. (2023)	Detect adversarial and evasion attacks targeting DL-based IDS models	Proposed a transfer-learning-based multi-detector framework that enhances robustness against adversarial perturbed traffic	Transfer learning detectors trained on selected feature subsets; adversarial attack generation and detection evaluated on standard IDS datasets
Sun et al. (2024)	Capture relational behaviors among hosts and flows using graph learning	Presented an end-to-end GNN-based IDS capable of modeling network topology and interaction patterns	Graph construction from host-flow interactions → GNN classifier; experiments and ablation studies on benchmark datasets

3. Proposed Methodology

The proposed Intrusion detection framework consists of end –to-end deep learning pipeline. Figure 1 demonstrates the five stages of the proposed framework.

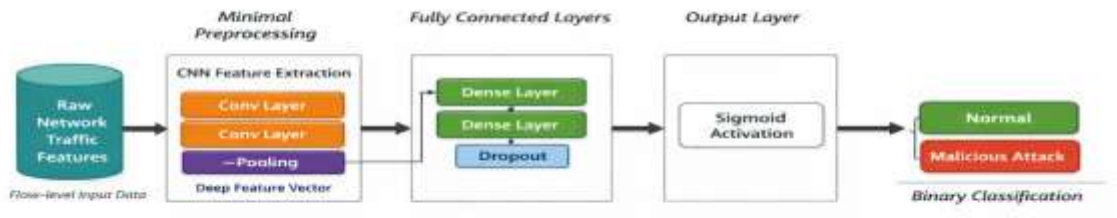

Figure 1: Proposed Framework

I. Raw traffic data
ii. Minimal Data cleaning
iii. CNN with batch normalization
Iv.Automatic feature learning
V.Attack classification

i.Raw Traffic data

The raw traffic data here it refers to the original flow based features which is provided in the dataset. Here no manual feature selection or handcraft feature transformation is required. The purpose of this approach is for ensuring that the learns directly from the realistic traffic data.

ii) Minimal Data Cleaning

The objective of this minimal data cleaning is to retail the original traffic characteristics by removing the attributes which are not contributing to detecting intrusions. The CSE-CIC –IDS2018 dataset contains several attributes that uniquely identify network flows but no decryption about traffic behaviour .The field removed are as follows

a. Flow ID
b..Timestamp
c.IP address
d. Port numbers

This removal of attributes is to prevent the data leakage and to ensure the model learns behaviour –based patterns rather than the network specific identifiers. Due to the traffic measurement limitations the missing values and positive or negative infinite values are replaced with 0.The batch normalization layers which is included in CNN avoids human intervention and supports end-to-end learning model .

iii. CNN based automatic feature learning

CNN is used as the core learning model. The best feature of this CNN is the convolutional layers automatically learn the spatial correlations among the heavy traffic features. It captures the interactions between both the statistical and temporal attributes .The steps involved in the CNN feature learning is illustrate din figure 2.

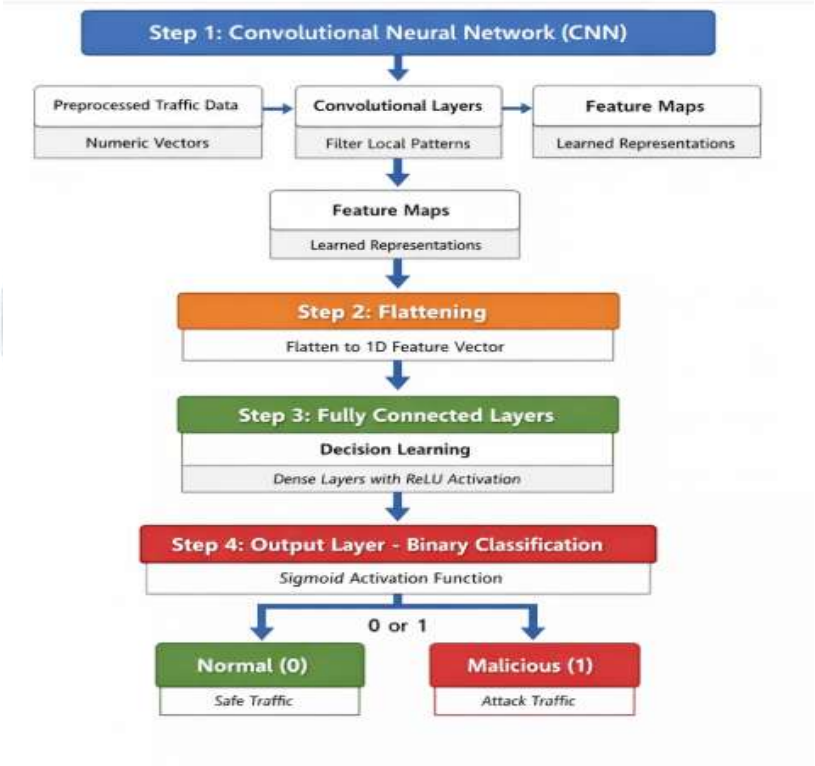

Figure 2: Steps involved in CNN

Step 1: Learning the deep features with CNN are as follows,

After minimal pre-processing each network traffic road is represented as a fixed-length numeric vector. This is reshaped and fed into the CNN.Covolutional layers slide filters over the input features. Each filter detects local patterns such as,

- sudden packet rate changes
- abnormal byte distributions
- Unusual flow duration combinations.

The CNN output feature maps which are automatically learned representations of the network behaviour.

Step 2: Flattening

The feature maps generated by the final convolution layer are still muti-dimensional.These are converted into a 1-D feature vector using a flatten operation. The following code is used for the flatten operation.

```
Feature maps (3D) → Flatten → 1D deep feature vector
```

This vector now represents high-level attack-aware features, learned directly from data.

Step 3: Fully Connected (Dense) Layers

To completely connect (Dense) layers, the flattened deep feature vector is provided. The fully connected (dense) layers learn global relationships among the convolutional layers' extracted features and integrate these patterns to construct an effective decision boundary distinction between benign and malignant traffic. The model may capture complicated data by using non-linear activation functions like ReLU and non-linear attack behaviours, thereby improving discrimination between benign and intrusive network activities.

Step 4: Output Layer – Binary Classification (Normal vs Malicious)

A Sigmoid activation function is used:

- Output range: 0 – 1
- Output interpretation:
  0 → Normal traffic
  1 → malicious traffic

Decision rule used in implementation is given below,

```
If output ≥ 0.5 → Malicious
Else → Normal
```

The high level feature representation learned by the CNN are passed sigmoid activation function, then completely connected layers. The final output classifies network traffic into NorAm or malicious categories. Each traffic flow must be classified as either normal or malicious, only one probability value is needed. Decision is threshold-based. into normal or malicious categories. Based on the output layer:

- Each incoming traffic flow is automatically classified
- The system labels traffic as:
  - Normal
  - or Malicious attack
  -

4. Expeimental setup

Each study was carried out using a typical computing platform to prove that the suggested intrusion detection system can be implemented without specialized hardware. Using the Python programming language, the implementation was completed. Using the Keras API and Tensor Flow, the deep learning models were developed and trained, executed on a system running Microsoft Windows.

The machine used for the tests included the following features:

- Programming Language: Python 3.x
- Processor: Intel Core i5 / i7
- Memory: 8 GB RAM or higher
- Operating System: Windows 10

No GPU acceleration was required for the baseline model, highlighting the lightweight nature of the proposed approach.

Dataset and Data Partitioning

The proposed intrusion detection system is evaluated using the CSE-CIC-IDS2018 benchmark dataset from the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE). The dataset is designed to reflect realistic enterprise network traffic and overcomes many limitations of earlier IDS datasets by including contemporary attack scenarios and diverse traffic patterns. CSE-CIC-IDS2018 was the dataset utilized for the testing. The dataset was divided between training and testing parts using an 80:20 split after some data cleaning. They used the training set to learn the model parameters and the testing set to evaluate the generalization performance.

Model Configuration

The proposed CNN-based IDS were configured with multiple one-dimensional convolutional layers to train discriminative traffic characteristics, followed by max-pooling and batch normalization layers. ReLU-activated fully connected layers were used for decision learning. In the output layer, binary classification was carried out using a Sigmoid activation function.

Training Parameters

The following hyperparameters were used for training the model:

- Optimizer: Adam
- Learning Rate: 0.001
- Loss Function: Binary Cross-Entropy
- Batch Size: 64
- Number of Epochs: 50

To enhance training stability to prevent overfitting, early interruption was used.

Evaluation Metrics

F1-Score, Accuracy, Precision, and Recall are popular metrics used to evaluate the performance of the proposed intrusion detection system. The detection capabilities are thoroughly evaluated using these metrics, particularly in distinguishing malicious traffic from benign network activity. The sample code used for evaluation metrices is shown in figure 3.

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

accuracy = accuracy_score(y, y_pred)
precision = precision_score(y, y_pred)
recall = recall_score(y, y_pred)
f1 = f1_score(y, y_pred)

print(f"Accuracy: {accuracy*100:.2f}%")
print(f"Precision: {precision*100:.2f}%")
print(f"Recall: {recall*100:.2f}%")
print(f"F1-score: {f1*100:.2f}%")
```

Figure 3 : sample code of metrics used in the code

1. Accuracy

Measures the overall correctness of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision

Indicates how many predicted attacks are actually malicious.

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

3. Recall (Detection Rate)

Measures how effectively the model detects malicious traffic.

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

4. F1-Score

Harmonic mean of precision and recall.

$$\text{F1-Score} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

6. Results

Table 2 shows the usual evaluation criteria that use a number of conventional models for machine learning to assess the effectiveness of the proposed CNN-based intrusion detection system. According to the comparative results, the CNN-based model surpasses conventional classifiers in terms of detection accuracy and resilience. Traditional machine learning approaches rely heavily on manually engineered features and fixed decision boundaries, which limits their ability to generalize to complex and evolving attack patterns. In contrast, the proposed CNN automatically learns hierarchical feature representations directly from network traffic data, enabling more effective discrimination between normal and malicious activities.

Experimental results show that the CNN-based IDS achieve an overall accuracy of 95.8%, which is significantly larger than the conventional models for machine learning. Furthermore, the deep learning approach exhibits improved detection capability for complex attack behaviors while maintaining a lower false alarm rate. The comparative analysis confirms that CNN-based feature learning provides superior representation power compared to handcrafted features used in machine learning techniques. Experimental comparisons show that CNN-based IDS outperform SVM and Random Forest are examples of classic machine learning models, consistent with findings reported in previous studies [1], [5]. These results validate the efficiency of deep learning in the identification of intrusions and highlight its suitability as a baseline model for more advanced and hybrid IDS frameworks.

Table 2: Traditional machine learning model comparison

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
SVM	92.6	92.1	91.8	91.9
Random Forest	94.3	94.0	93.6	93.8
Proposed CNN (Solution-1)	**95.8**	**95.4**	**96.1**	**95.7**

7. Conclusion

This study used raw network traffic data utilizing the CSE-CIC-IDS2018 dataset to provide a comprehensive intrusion detection system based on deep learning. By applying minimal preprocessing and leveraging CNN-based automatic feature learning, the proposed IDS achieve an accuracy of 95.8%, outperforming traditional machine learning approaches. The experimental results confirm the models used for deep learning work well with intrusion detection tasks and can effectively learn discriminative representations directly from network traffic data. This work establishes a strong baseline and demonstrates the effectiveness of CNN-based systems for automatically classifying attacks and extracting features. Future enhancements of this work may extend the proposed framework by incorporating more advanced deep learning architectures and intelligent optimization strategies. Latent feature learning models can be explored to improve representation quality, reduce redundancy, and address data imbalance issues, particularly for complex and low-frequency attack patterns. Future work may explore latent feature learning models and hybrid deep learning architectures to improve detection of complex and low-frequency attacks [6], [7], [9], [12].In addition, hybrid deep learning frameworks that jointly capture spatial patterns, temporal dependencies, and relational interactions within network traffic can be investigated to better model evolving and coordinated attack behaviors. It is anticipated that these improvements would increase detection accuracy, resilience, and adaptability further, making the intrusion detection system more effective for large-scale network systems in the real world.

References

[1] J. Kim, Y. Shin and E. Choi, “An Intrusion Detection Model based on a Convolution Neural Network,” J. Multimedia Inf. Syst., vol. 6, no. 4, pp. 165–172, 2019.
[2] A. B. Bouidaine, D. Moussaoui, M. Hadjila, W. Ferhi and M. H. Hachemi, “Deep Learning-Based Anomaly and Intrusion Detection Using the CSE-CIC-IDS2018 Dataset,” Eng. Technol. Appl. Sci. Res., vol. 15, no. 4, pp. 24782–24787, 2025.
[3] Y. C. Wang, Y. C. Houng, H. X. Chen and S. M. Tseng, “Network Anomaly Intrusion Detection Based on Deep Learning Approach,” Sensors, vol. 23, no. 4, 2023, doi:10.3390/s23242171.
[4] A. Dey, “Deep IDS: A Deep Learning Approach for Intrusion Detection Based on IDS2018,” in 2020 2nd Int. Conf. Sustainable Technologies for Industry 4.0 (STI), 2020.
[5] A. B. Farhan and A. D. Jasim, “Performance Analysis of Intrusion Detection for Deep Learning Model Based on CSE-CIC-IDS2018 Dataset,” Indones. J. Electr. Eng. Comput. Sci., vol. 26, no. 2, pp. 1165–1172, 2022.
[6] A. Chakrawarti and S. S. Shrivastava, “Intrusion Detection System using LSTM and Fully Connected Neural Networks on KDDCup99 and NSL-KDD Dataset,” Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 9s, pp. 621–635, 2023.
[7] R. Kale, Z. Lu, K. W. Fok and V. L. L. Thing, “A Hybrid Deep Learning Anomaly Detection Framework for Intrusion Detection,”, 2022.
[8] V. Imanbayev et al., “Collaborative Feature Maps of Networks and Hosts for AI-Driven Intrusion Detection,” in Proc. IEEE Global Communications Conference (GLOBECOM), 2022.
[9] Y. Zeng, “CSAGC-IDS: A Dual-Module Deep Learning Network Intrusion Detection Model for Complex and Imbalanced Data,”, 2025.
[10] “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,” Appl. Sci., vol. 9, no. 20, 2019, discussing KDDCup99, NSL-KDD, UNSW-NB15 datasets.
[11] “Deep Learning Enabled Intrusion Detection System for IoT Security,” J. Wireless Commun. Netw., 2025, comparing performance on NF-UNSW-NB15 and NF-CSE-CIC-IDS2018 datasets.
[12] S. Smith and P. Patel, “A Hybrid BiLSTM-CNN Approach for Intrusion Detection Using UNSW-NB15 and NSL-KDD Datasets,” Scientific Reports, 2025, pp. 1–12.