International Journal of Synergy in Engineering and Technology
IJSET
Journal homepage: https://tatiuc.edu.my/ijset/index.php/ijset/ e-ISSN: 2735-0622

Development of a CNN-Based IDS Model for Network Traffic Classification

Wan Mohamed, W. A. A.*, Rusli, S. N. S., Mohd Razali, S. A.

University College TATI, Jalan Panchor, Telok Kalong, 24000 Kemaman, MALAYSIA

*Corresponding author: ainulalyani@uctati.edu.my

KEYWORDS

Intrusion Detection System (IDS)
Cybersecurity
Convolutional Neural Network (CNN)
Machine Learning
Network Threat Detection
Flask Web Application

ABSTRACT

Cybersecurity threats are becoming increasingly frequent and complex as organizations rely more on interconnected digital networks. Traditional security methods often struggle to detect advanced and unknown attacks, such as zero-day exploits. Intrusion Detection Systems (IDS) are essential for monitoring network activity and identifying malicious behavior; however, conventional signature-based IDS are limited in detecting evolving threats. This paper presents the design and development of an Intelligent IDS web application that integrates a Convolutional Neural Network (CNN)-based detection model to enhance intrusion detection capability and system usability for network traffic analysis. The focus of this study is on system implementation, workflow integration, and visualization of detection outputs rather than performance optimization. The system analyzes network traffic data to classify activities as normal or malicious using a trained CNN model. The development process follows a structured methodology, including system design, model integration, and web-based deployment using the Flask framework. A dashboard interface is implemented to visualize detection results, enabling users to monitor network activity in real time. The system output demonstrates the capability to process network traffic data and present categorized intrusion detection results through an interactive and accessible platform. The proposed Intelligent IDS provides a practical tool for cybersecurity monitoring and contributes to enhancing understanding of AI-based intrusion detection systems in modern network environments.

Received 11 January 2026; Revised 25 February 2026; Accepted 31 March 2026; Published 01 April 2026

1.0 INTRODUCTION

The rapid growth of digital technologies and network connectivity has greatly increased the vulnerability of computer systems to cybersecurity threats. Modern organizations depend on network infrastructures to manage communication, data exchange, and service delivery. However, this reliance also creates opportunities for cyber attackers to exploit weaknesses in networks and information systems. Attacks such as Distributed Denial-of-Service (DDoS), brute-force login attempts, and reconnaissance methods like port scanning have become more frequent and

sophisticated [1,2]. IDS plays a critical role in protecting network infrastructures by monitoring network traffic and identifying suspicious activities. Traditional IDS typically rely on signature-based detection techniques, where incoming network traffic is compared with a database of known attack patterns. While this approach is effective in detecting previously identified threats, it struggles to recognize unknown or evolving attack patterns [3]. As cyber threats become more complex, traditional IDS frameworks face challenges related to detection effectiveness, scalability, and false alarm rates [4].

To overcome these limitations, researchers have explored the integration of Artificial Intelligence (AI) and Machine Learning (ML) techniques in IDS. AI-based IDS can analyze large volumes of network data and learn patterns associated with malicious activities, enabling the detection of both known and unknown attacks more effectively [5,6]. Among deep learning models CNNs have demonstrated strong performance in identifying complex patterns within high-dimensional datasets, making them suitable for network intrusion detection [7].

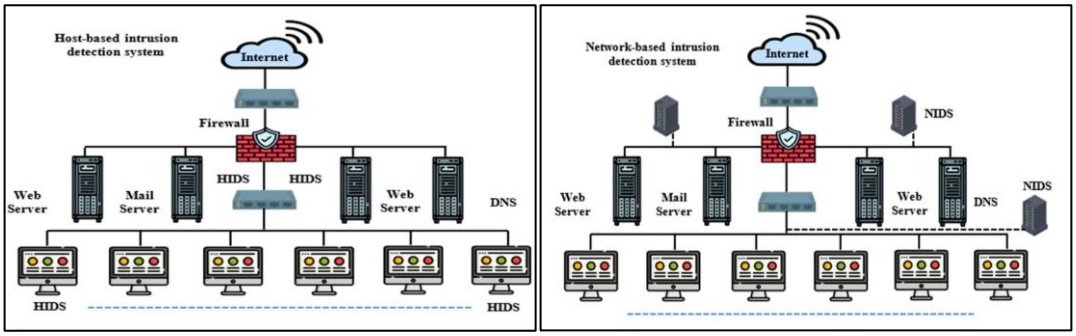
The primary objective of this study is to design, develop, and deploy a CNN-based Intelligent IDS web application that enables users to classify network traffic and visualize potential threats. The web application is implemented using Flask, a lightweight Python framework suitable for building web interfaces and efficiently integrating machine learning models. NSL-KDD and CICIDS2017 were selected to provide complementary coverage of classic and modern network attack scenarios, ensuring the CNN model is trained and validated on diverse traffic patterns. The Software Development Life Cycle (SDLC) methodology was employed to ensure systematic planning, implementation, and testing, making it appropriate for developing the prototype system in a structured manner.

The main contributions of this study include the design and development of a CNN-based IDS model for network traffic classification, integration of the model into a Flask-based web application, implementation of a visualization dashboard for interpreting detection outputs, and demonstration of a functional IDS workflow for practical cybersecurity monitoring and educational purposes.

2.0 LITERATURE REVIEW

2.1 Intrusion Detection Systems

IDS are security mechanisms designed to monitor network traffic and detect malicious activities within a system. IDS solutions analyze network behavior and generate alerts when suspicious patterns or unauthorized access attempts are detected [8]. IDS technologies are generally classified into two major categories: Network-Based Intrusion Detection Systems (NIDS) and Host-Based Intrusion Detection Systems (HIDS) [9]. Figure 1(a) shows NIDS monitor network traffic across an entire infrastructure by inspecting packets transmitted through network devices such as routers and switches. Figure 1(b) shows HIDS focus on monitoring activities within individual systems by analyzing system logs, file integrity, and application behavior [10]. Both approaches provide valuable insights into potential security breaches, although they operate at different levels of the network architecture.



(a) (b)

Figure 1: Architecture of IDS: (a) HIDS; (b) NIDS [10]

Detection techniques used in IDS typically include signature-based detection, anomaly-based detection, and hybrid approaches. Signature-based detection identifies attacks by matching traffic patterns against a database of known attack signatures. Although this method provides high accuracy for known threats, it is ineffective against new or unknown attacks [11]. Anomaly-based detection, on the other hand, establishes a baseline of normal network behavior and identifies deviations that may indicate malicious activity. Hybrid detection approaches combine both methods to enhance detection accuracy and reduce false positives [12].

2.2 Cyberattacks and Detection Challenges

Cyberattacks targeting network infrastructures have increased significantly in recent years. Among the most common attack types are DoS attacks, brute-force authentication attempts, and reconnaissance activities such as port scanning [13].

DoS attacks aim to overwhelm network resources by flooding systems with excessive traffic, thereby preventing legitimate users from accessing services [14]. Brute-force attacks attempt to gain unauthorized access by systematically trying multiple username-password combinations until the correct credentials are discovered [15]. Port scanning is a reconnaissance technique used by attackers to identify open network ports and potential vulnerabilities before launching further attacks [16].

Detecting such attacks is challenging due to the high volume of network traffic and the increasing use of encryption, which can obscure malicious patterns within data streams. These challenges highlight the need for intelligent detection systems capable of analyzing complex network behavior [17].

2.3 AI in Intrusion Detection

AI and ML techniques have emerged as powerful tools for enhancing intrusion detection capabilities. ML algorithms can analyze large datasets and automatically learn patterns associated with malicious behavior. These techniques allow IDS to detect anomalies and previously unseen attacks more effectively than traditional rule-based systems [18]. Figure 2 shows the taxonomy of machine learning, illustrating the main categories and common algorithms used in IDS.

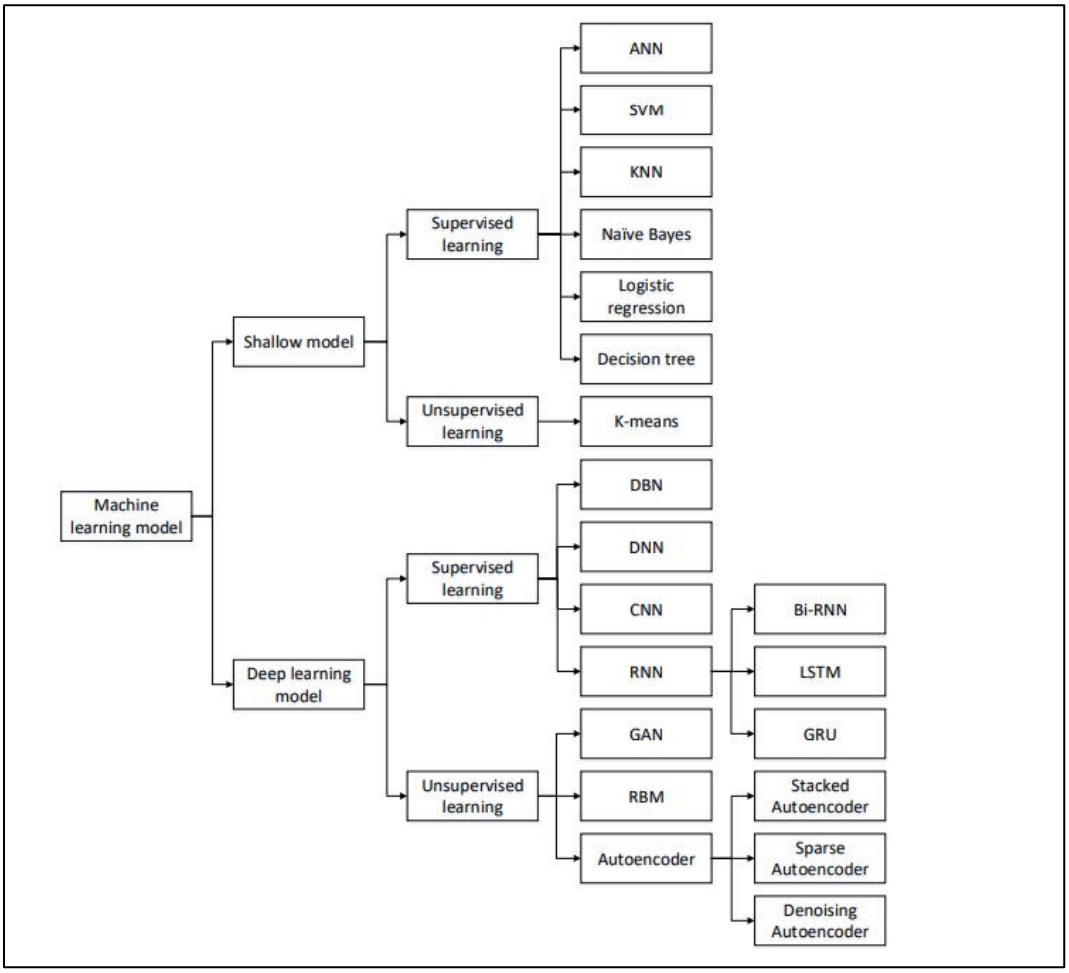


Figure 2: Taxonomy of machine learning algorithms [21]

Common machine learning algorithms used in IDS include Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Trees (DT), and Random Forest models [19]. These models classify network traffic based on extracted features and can detect patterns indicative of cyberattacks.

Deep learning techniques have further improved intrusion detection performance by enabling hierarchical feature extraction. Models such as CNN, Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks have shown promising results in cybersecurity applications [20].

CNN models are particularly effective for analyzing structured datasets because they can automatically extract spatial features from input data. In IDS applications, network traffic features can be transformed into matrix representations that allow CNN models to identify patterns associated with attack behaviours [21]. Figure 3 illustrates the structure of the CNN model used in this study, which processes the input feature matrix through multiple convolutional and pooling layers to learn hierarchical patterns before classification.

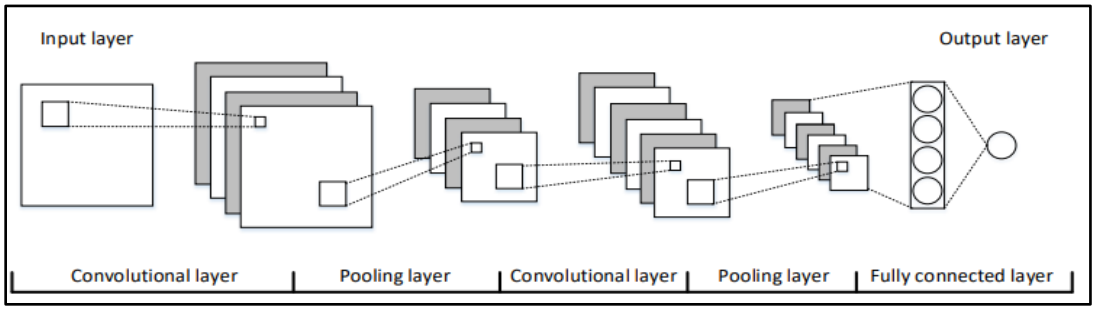


Figure 3: Proposed CNN architecture consisting of convolutional, pooling, and fully connected layers for network traffic classification [21].

3.0 METHODOLOGY

3.1 IDS Development Approach

This study adopts the SDLC methodology to guide the development of the Intelligent IDS. SDLC provides a structured framework consisting of several stages, including planning, analysis, design, implementation, testing, and maintenance. The methodology ensures that system development is conducted systematically while maintaining software quality and minimizing development risks.

During the planning stage, the objectives, scope, development tools, and datasets required for the system are identified. The analysis phase focuses on understanding system requirements, including the intrusion detection workflow, attack classification categories, and functional features such as file upload, traffic analysis, and result visualization. The design phase then defines the overall system architecture, including the machine learning model structure and the web-based interface used to interact with the IDS.

3.2 IDS Architecture and Implementation

The Intelligent IDS is developed using Python-based technologies, integrating machine learning capabilities with a web-based interface. The detection engine utilizes a CNN model to classify network traffic into normal and malicious categories. Prior to model training, network traffic datasets undergo pre-processing steps such as data cleaning, normalization, and feature preparation.

The IDS is integrated with a Flask-based web application that allows users to upload network traffic data in CSV format. Once uploaded, the system pre-processes the data and feeds it into the trained CNN model for classification. The detection results are then presented through a visualization dashboard, which displays traffic summaries, detected threats, and prediction results. This architecture enables users to easily analyse network activities and identify potential security threats.

3.3 IDS Workflow

The operational workflow of the IDS begins when a user accesses the web application and uploads a network traffic file. The system first validates the uploaded file before performing data pre-processing. The processed data is then analysed using the trained CNN model, which classifies the network traffic into categories such as normal activity or potential intrusion attacks.

The results of the analysis are displayed on a dashboard that includes charts, tables, and traffic summaries to assist users in monitoring network behaviour. Users can also visualize traffic patterns and review recent detection results to better understand potential security threats.

The workflow of the proposed IDS is illustrated in Figure 4, which demonstrates the process from data upload to CNN-based traffic classification and dashboard visualization. The process begins when

a user accesses the web application interface and uploads a network traffic file, typically in CSV format. The system first performs a file validation process to ensure that the uploaded data meets the required format and structure. If the file is invalid, the system returns an error message and prompts the user to upload a valid file.

Once the file passes the validation stage, the system proceeds to the data pre-processing phase, where the dataset undergoes cleaning, normalization, and feature preparation to ensure compatibility with the machine learning model. After pre-processing, the prepared dataset is forwarded to the CNN detection model, which performs classification to determine whether the network traffic represents normal activity or a potential cyberattack.

The CNN model analyses traffic patterns and categorizes them into several classes, including normal traffic, DoS attacks, brute force attacks, port scanning activities, or other intrusion attempts. Following classification, the system generates detection results and sends them to the visualization dashboard.

The dashboard presents the analysis results through traffic summaries, graphical charts, and tables of detected threats, allowing users to easily interpret the network behaviour and identify potential security incidents. After reviewing the results, users may choose to upload another dataset for further analysis, or terminate the process.

This workflow ensures a structured and automated intrusion detection process, enabling efficient analysis of network traffic while providing clear visual feedback for cybersecurity monitoring.

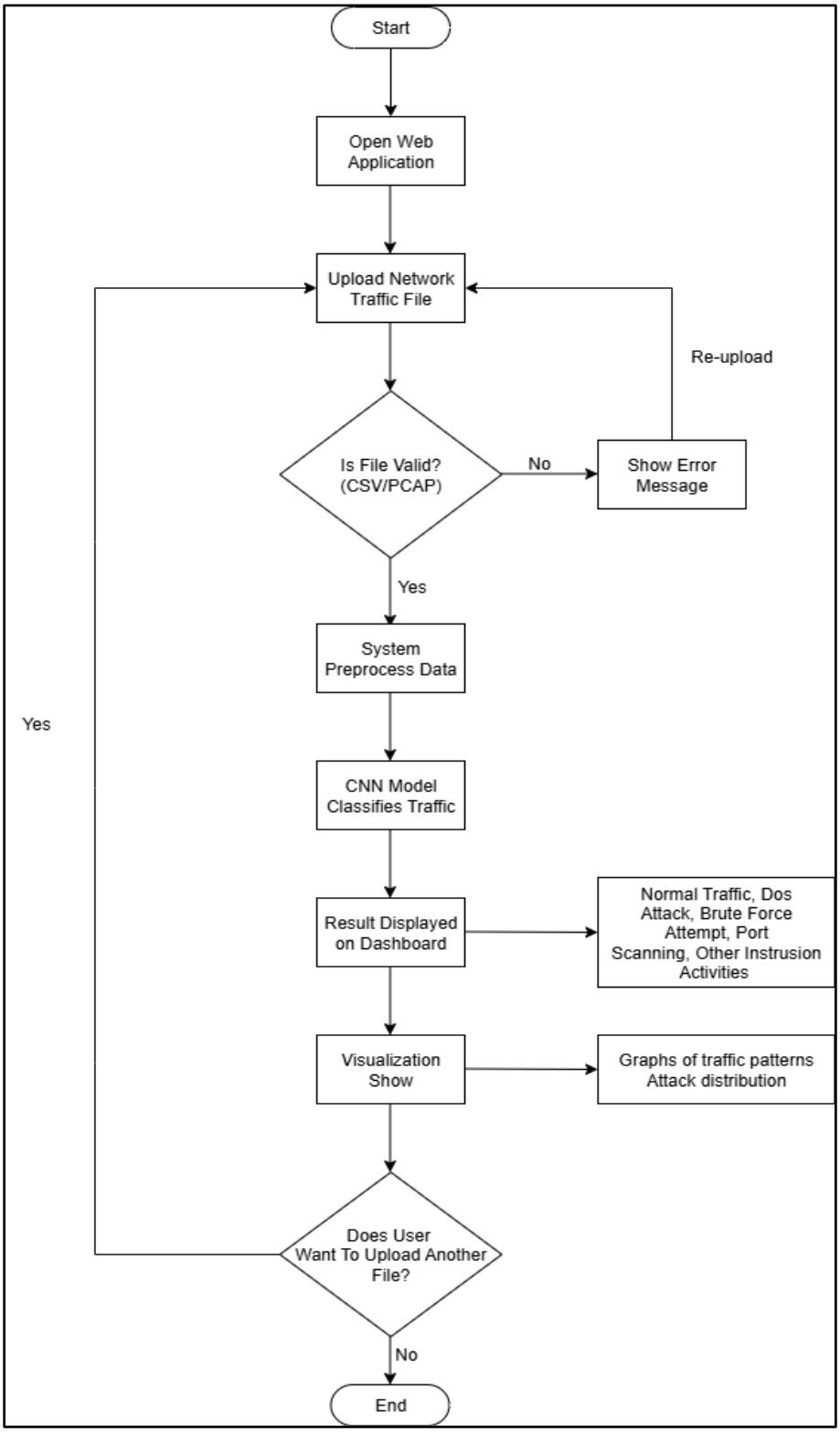


Figure 4: Workflow of the proposed Intelligent IDS.

4.0 SYSTEM IMPLEMENTATION AND OUTPUT

This section presents the implementation of the Intelligent IDS and evaluates its functional capabilities based on system outputs, user interaction, and visualization features. The objective is to demonstrate how the developed system processes network traffic data and presents intrusion detection results in an accessible manner.

The main user interface in Figure 5 provides an overview of the system functionalities, including dataset upload, analysis initiation, and real-time feedback. Users can select a dataset using the Choose File button and initiate the analysis by clicking Upload & Analyze. A progress indicator displays the current status of the processing, enhancing user experience and system transparency.

The IDS is designed to accept structured CSV files containing network traffic features suitable for CNN-based classification. The system first validates the uploaded dataset to ensure it follows the required format. If the file is unstructured or incomplete, an error message is displayed, prompting the user to upload a valid dataset. This validation ensures reliable processing and consistent reporting.

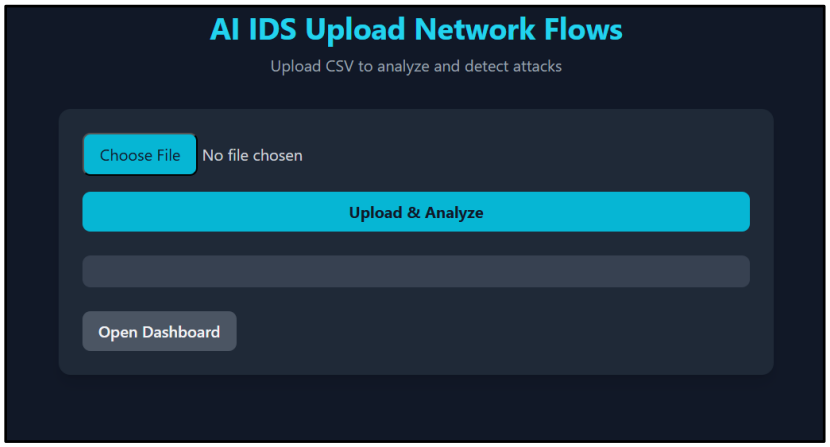


Figure 5: Main user interface of the Intelligent IDS, showing dataset upload and analysis features.

The dashboard interface provides analytical insights into network traffic behavior through three main reporting components:

1. *Traffic Summary Visualization* – Displays the proportion of different traffic categories, allowing users to quickly assess overall network conditions.
2. *Flagged Flows Table* – Highlights potentially malicious activities, enabling users to investigate specific threats in detail.
3. *Recent Predictions Panel* – Shows real-time classification results generated by the CNN model, demonstrating the system's dynamic processing capability

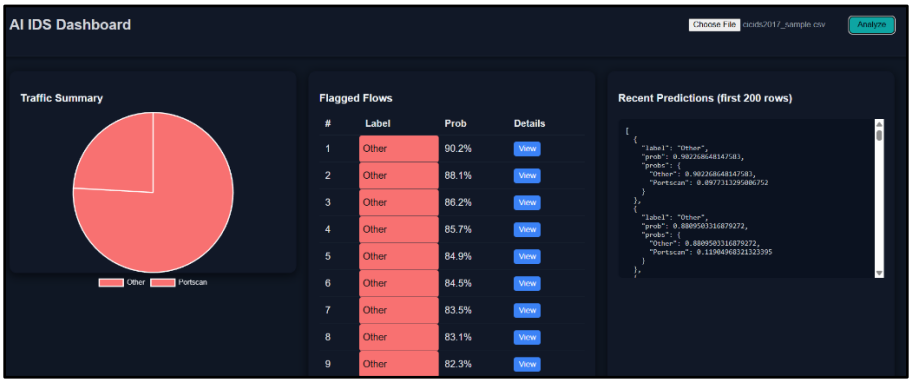


Figure 6: Dashboard interface of the Intelligent IDS, displaying traffic summary, flagged flows, and prediction outputs.

The system outputs demonstrate that the CNN model effectively classifies network traffic into normal and malicious categories, with detection of attack types such as DoS, brute-force, and port scanning. Functional validation was conducted to ensure the system operates as intended. The IDS successfully processed multiple network traffic datasets in CSV format, performed data pre-processing, and generated classification outputs without system failure.

The integration of the CNN model with the Flask-based web interface functioned smoothly, demonstrating the system's reliability for practical use. The dashboard visualization effectively presents the detection results, supporting user interpretation of network traffic behavior and facilitating informed cybersecurity monitoring.

5.0 CONCLUSION

This study successfully developed and implemented an Intelligent IDS web application that integrates a CNN-based detection model with an interactive visualization dashboard. The system demonstrates the feasibility integrating machine learning models within web-based platforms for network traffic classification.

The developed IDS provides practical value for cybersecurity practitioners by offering a tool for monitoring, analyzing, and visualizing network traffic in an accessible manner. Its dashboard enables users to quickly identify potential threats, facilitating informed decision-making in network security management. From an innovation perspective, the integration of CNN-based detection with a web interface represents a prototype that bridges AI research and practical cybersecurity applications.

For future work, the prototype can be enhanced and deployed in real-world scenarios as a HIDS or NIDS, enabling continuous monitoring of enterprise networks. Additionally, expanding support for multiple dataset formats and automated threat reporting could further improve usability and adoption in operational environments.

Author Contribution

Wan Mohamed, W. A. A.: Investigation, supervision, writing, and editing. Rusli, S. N. S.: Development, methodology, visualisation, writing and editing. Mohd Razali, S. A: Supervision, development and visualization.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems. IEEE Access. 2019;7:191–201.
[2] Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor. 2016;18(2):1153-1176.
[3] Sommer R, Paxson V. Outside the closed world: on using machine learning for network intrusion detection. IEEE S&P. 2010.
[4] Valdovinos IA, et al. A survey of machine learning methods for intrusion detection. Comput Netw. 2021.
[5] Shone N, Ngoc TN, Phai VD, Shi Q. Deep learning approach for network intrusion detection. IEEE Trans Emerg Topics Comput Intell. 2018.
[6] Kim G, Lee S, Kim S. A novel hybrid intrusion detection method integrating anomaly detection. Expert Syst Appl. 2020.
[7] Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems. Appl Sci. 2019.
[8] Scarfone K, Mell P. Guide to intrusion detection and prevention systems. NIST. 2018.
[9] Liao H, Lin CHR, Lin Y, Tung K. Intrusion detection system: a comprehensive review. J Netw Comput Appl. 2013.
[10] Axelsson S. Intrusion detection systems: a survey. Tech Rep. 2018.
[11] Behl A, Behl K. Cybersecurity and cyberwar. Springer. 2017.
[12] Garcia-Teodoro P, et al. Anomaly-based network intrusion detection. Comput Secur. 2009.
[13] Mahesh D, Kumar TS. Detection of DDoS attacks using ML. 2024.
[14] Alshammari A, Aldribi A. Machine learning based intrusion detection. 2021.
[15] Park J, Kim J, Gupta B, Park N. Brute-force attack detection methods. 2021.
[16] Everson D, Cheng L. Port scanning detection using ML. 2024.
[17] Salem O, et al. Network intrusion detection challenges. 2024.
[18] Katiyar N, et al. AI-based cybersecurity intrusion detection systems. 2024.
[19] Maseer ZK, et al. Comprehensive survey on ML IDS. IEEE Access. 2021.
[20] Yin C, Zhu Y, Fei J, He X. Deep learning approach for intrusion detection using RNN. IEEE Access. 2017.
[21] Vinayakumar R, et al. Deep learning for IDS: CNN model. IEEE Access. 2019.
[22] Tharwat A. Classification assessment methods. 2020.
[23] Mugiraneza C. False positive reduction in IDS. 2025.
[24] Nobakht M, et al. Deep learning approaches for cyber security. 2016.
[25] Jagadish H, et al. Big data and privacy challenges. 2014.
[26] Otoum S, Nayak A. Hybrid deep learning intrusion detection. 2021.
[27] Kim H, et al. Adaptive AI intrusion detection systems. 2020.
[28] Thirimanne S, et al. Evaluation of NSL-KDD dataset. 2022.
[29] Sharafaldin I, Lashkari A, Ghorbani A. Toward generating realistic IDS datasets (CICIDS2017). 2018.

