



A Multi-Class Web Intrusion Detection System Using Deep Learning

Hiba Allah Ziyad^{1*}  , and Amer Abdulmajeed Abdulrahman²  
^{1,2} Computer Science Department, College of Science, University of Baghdad, Baghdad, Iraq

*Corresponding Author.

Received:22/February/2026
Accepted:29/June/2026
Published: 20/October/2026
<https://doi.org/10.30526/39.4.4466>



© 2026. The Author(s). Published by College of Education for Pure Science (Ibn Al-Haitham), University of Baghdad. This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Abstract

Web applications are an integral part of digital infrastructure in the modern era and have become increasingly susceptible to advanced cyber threats, thus needing more intelligent and adaptable intrusion detection system. Traditional signature and rule-based (IDS) systems have inherent limitations in detecting various types of web attacks because they are static in nature. This paper presents a multi-class web intrusion detection framework based on deep learning methods by using the WEB-IDS23 dataset, a realistic representation of modern web traffic environments and presents the first holistic multi-class evaluation on this dataset by employing an integrated deep learning and stacking-based ensemble model. The model utilizes Random Forest-based feature selection and combines it with three deep learning models (One-Dimensional Convolutional Neural Network (1D-CNN), Deep Neural Network (DNN) and Gated Recurrent Unit (GRU-RNN)). we then used a stacking-based meta-learning approach to aggregate the predictions. Strong detection performance is demonstrated in experiments. The 1D-CNN yielded best standalone accuracy of 96.43% and F1-score of 96.32%, while stacking ensemble further boosting the classification accuracy to 96.55% and precision to 97.15%. Although the stacking meta-model achieved a marginally lower F1-score compared to its standalone 1D-CNN, it performed with higher overall accuracy. This phenomenon is due to the imbalanced class distribution of the WEB-IDS23 dataset, where optimizing majority-class classification raised overall accuracy but lowered recall and precision on some minority classes. By combining these methods with feature selection, deep learning and stacking-based ensemble learning, the final result achieves higher accuracy in detection task as well as robust detection capability and scalability. The proposed system is an efficient approach for the multi-class web intrusion detection system.

Keywords: Web Intrusion Detection, Deep Learning, Random Forest Feature Selection, Stacking Ensemble Learning, Multi- Class Classification.

1. Introduction

Web applications are an integral part of today's digital infrastructure on which critical functions in government, industry, commerce, and finance sectors rely. Their persistent accessibility to the open web has exponentially increased their attack exposure surface, exposing them to complex cyber threats like SQL injection (SQLi), cross-site scripting (XSS), distributed denial-of-service (DDoS) and other web application attacks¹⁻⁴. The continuing evolution of these threats threatens data integrity, service availability, and business continuity. Traditional signature-based and rule-based intrusion detection systems (IDSs) are dependent on predefined attack patterns which are manually designed. While such systems have been successful in dealing with the threats, they lack of detecting newly issued and polymorphic attacks occasionally, because of their static nature and limited generalization ability⁵⁻⁷.

Therefore, there is a growing need for smart and adaptive intrusion detection systems to tackle dynamic web threats. In recent years, deep learning technologies have greatly improved the detection accuracy of attacks. CNNs are particularly useful in capturing discriminative spatial features for network traffic^{8,9}, while RNNs and GRUs have been shown to be highly effective architectures in modeling temporal dependencies within a sequence of traffic flows¹⁰. Hybrid models comprising convolutional and recurrent elements have also enhanced the robustness of detection as well as accuracy of classification^{11,12}.

Furthermore, there is evidence from systematic studies that deep learning intrusion detection systems outperform traditional machine-learning methods in high-dimensional and complex traffics data¹³. Still there exist several problems even with these improvements. Datasets for intrusion detection are often class imbalanced, and this imposes a bias on models in favor of majority classes. Detection capabilities for minority categories of attacks suffer as a result¹⁴. Furthermore, numerous prior studies are based on outdated or non-web-centric datasets that may not be indicative of internet attack behaviors in the present¹⁵.

This paper addresses these issues using the recently proposed WEB-IDS23 dataset that emulates realistic web traffic, and a variety of modern web attacks¹⁶. In the proposed framework, we combine Random Forest-based feature selection and three deep learning models including One-Dimensional Convolutional Neural Network (1D-CNN), Deep Neural Network (DNN) and Gated Recurrent Unit (GRU-RNN).

To strengthen the multi-class web intrusion detection performance in uncertain scene, a stacking-based meta-learning approach is adopted to ensemble probabilistic outputs of those individual models and mitigate the multi-class problem of imbalanced cases¹⁷. To the best of our knowledge, no existing work has ever performed a complete multi-class evaluation through a combined deep learning and stacking framework on WEB-IDS23. This work addresses this gap with the first systematic analysis of such architectures on this benchmark.

The main contributions of this study are summarized as follows:

1. Analyzing the effectiveness of novel WEB-IDS23 dataset for multi-class web intrusion detection.
2. Utilizing feature selection based on Random Forest to reduce the dimension of the data and enhance computational efficiency.
3. Comparing the performance of three deep learning models, namely 1D-CNN, DNN, and GRU-RNN, for web attack classification.
4. Developing Stacking-based meta-model to improve the overall detection performance on a web intrusion imbalanced dataset.

The remainder of this paper is structured as follows: Section 2 presents the Materials and Methods; Section 3 describes the Results; Section 4 provides the Discussion; and Section 5 concludes the paper.

2. Materials and Methods

The Materials and Methods section presents the WEB-IDS23 dataset, the applied data preprocessing procedures, and the deep learning models used for multi-class web intrusion detection.

2.1 WEB-IDS23 Dataset

We use the WEB-IDS23 dataset as the main benchmark for evaluating the proposed web-based intrusion detection system¹⁶. WEB-IDS23 is a new dataset created to emulate real web traffic and contemporary HTTP/HTTPS attacks. In contrast with previous IDS datasets that are generic network traffic, the dataset is specific for the threats on web application and represents a similar attack nature as found in modern deployment scenarios. The original dataset contains around 12 million network flows represented by 38 traffic features.

These features provide statistical and protocol-level properties of the web traffic, including flow duration statistics, packet level attributes, flag-based metrics, payload-related attributes and

window-based features. The web attack categories covered by the dataset include brute force (HTTP/HTTPS), SQL injection (HTTP/HTTPS), DoS, XSS, SSRF, reverse shell attacks, FTP login attempts and SSH logins events, port scanning as well as sweep host activity and SMTP-related attacks^{1,3}.

One of the main reasons for choosing WEB-IDS23 is that it includes a wider variety of modern and evolved web-based attack types, compared to previous benchmark datasets¹⁶. Such diversity allows for more thorough analysis of multi-class intrusion detection systems in web scenarios¹⁸. The dataset is highly imbalanced with some types of attack dominating the traffic distribution and others occurring much less frequently. Class imbalance is known to be one of the main difficulties for intrusion detection modeling¹⁴.

A large number of duplicated and redundant traffic records were discovered and removed to eliminate bias and improve training efficiency during the preprocessing stage., two million traffic instances were kept for experimental evaluation.

2.2 Data Preprocessing

In this study using the WEB-IDS23 dataset, we performed several preprocessing techniques in order to make it suitable for training with high performance of accuracy, stability and computational time of deep learning models.

Data preprocessing is a key process in any intrusion detection system, since it improves data quality, reduces redundancy and helps the learning algorithms to have good generalization performance¹⁹. The preprocessing applied in this study involves the data merging, data cleaning, categorical encoding, feature scaling, feature selection and stratified splitting of a dataset.

2.2.1 Data Cleaning and Merging

Data cleaning is the process of removing irrelevant attributes, duplicate records, and inconsistencies that may hamper model learning and evaluation. In big network traffic datasets, redundant flows could result in bias and overfitting. First, all CSV files of the WEB-IDS23 dataset were merged into a unified Dataframe.

The raw dataset with 38 feature attributes described around 12 million traffic instances. Informative identifier-based attributes (uid, ts, id.orig_h, id.resp_h, service and attack) were deleted because they don't help in the classification of the attack. The attack column was excluded in preprocessing because it was not used as an input feature or target label for the framework proposed.

The actual target label for the multi-class classification was attack_type. After, duplicate traffic samples were detected and removed in a manner that prevented repeats from distorting the learning process. Removing all duplicates leaves about 2 million valid traffic instances in the dataset. There were also no missing values identified in the preprocessed data, which results in consistency of data prior to model construction.

2.2.2 Encoding of Categorical Variables

Deep learning models require numeric input and the categorical features need to be encoded into numbers. In this work, the categorical features were encoded by Label Encoder technique. We use Label Encoding rather than One-Hot Encoding so as not to increase memory usage and computational complexity, since the WEB-IDS23 dataset contains a high volume of traffic instances. Appropriate encoding methods increase efficiency without dimensionality addition²⁰.

2.2.3 Feature Scaling (Standardization)

Feature scaling is necessary for successful deep learning to enable equally participation of numerical features in the process of training. Large numeric ranges of features, if not normalized will make some dimensions with large numeric values influence gradient updates far more than others and can cause instabilities in the convergence rate²⁰.

Standard Z-score normalization was performed based on the Standard Scaler functionality due to which each feature had zero mean and unit variance. The stability of convergence and the robustness of training model in IDS systems can be increased by appropriate normalization. To avoid information leakage, the scaling parameters were computed on training data and

transformed on the testing subset.

2.2.4 Feature Selection

Feature selection aims to keep the informative attributes but discard irrelevant or redundant features, which helps achieving better computational efficiency and avoiding overfitting. High-dimensional effects among network traffic features can induce noise and overhead. Feature importance scores were computed using a Random Forest classifier with 200 estimators. The selection threshold was the mean feature importance value, using the Select From Model method.

Random Forest- based feature selection algorithms have been showed effective to enhance the detection performance of IDSs, and at the same time preserve discriminative traffic attributes^{21,22}. Features with importance scores higher than the threshold were then preserved. This step reduced the number of features to 16 significant attributes and increased the computational efficiency, while still being able to achieve classification.

Table1. Selected features after Random Forest-based feature selection

No.	Selected Feature
1	flow_duration
2	fwd_pkts_tot
3	bwd_pkts_tot
4	fwd_data_pkts_tot
5	bwd_data_pkts_tot
6	fwd_pkts_per_sec
7	bwd_pkts_per_sec
8	flow_pkts_per_sec
9	down_up_ratio
10	fwd_header_size_tot
11	bwd_header_size_tot
12	fwd_PSH_flag_count
13	bwd_PSH_flag_count
14	flow_ACK_flag_count
15	payload_bytes_per_second
16	fwd_last_window_size

2.2.5 Dataset Splitting and Class Imbalance Consideration

To obtain robust model evaluation in the context of imbalanced attack distributions, we split the preprocessed data into training and testing sets through stratified sampling. Stratified splitting maintains the initial distribution of classes in both subsets, which is very important in multi-class intrusion detection work. This ensured that the model evaluation was performed while preserving the original class distribution in both the training and testing datasets.

A 70%–30% split was used with fixed random state for reproducibility. No resampling methods were used, as the goal of this work was to assess the robustness of the models against realistic traffic imbalance and not artificial balanced data distributions. In addition, the dataset contained enough samples for most attack categories that models can learn representative traffic patterns without performing any manual balancing.

2.3 The Deep Learning Models

After preprocessing and feature selection, the reduced feature matrix with the 17 selected features was applied to train and evaluate different deep learning models in multi-class web intrusion detection. Deep learning methods have proven to be effective in capturing complex, non-linear and high-dimensional traffic representations for intrusion detection systems (IDSs)²³. Three deep learning architectures, namely 1D-CNN, DNN, and GRU-RNN, were implemented using TensorFlow/Keras with same training settings.

The AdamW optimizer was used with a learning rate of 1×10^{-4} and weight decay of 1×10^{-5} , because adaptive optimization methods are reported to lead stable converge and generalization on security related tasks. We used sparse categorical cross-entropy as the loss function because

class labels were encoded in integers. We used early stopping and learning rate reduction on plateau to avoid overfitting and stabilize the training.

To enable stable convergence and mitigate overfitting, the hyperparameters of the implemented models such as number of layers, neurons, dropout rates were selected empirically through multiple trainings with various combinations followed by performance comparisons. The final hyperparameter configuration was selected based on validation performance and training stability.

The 1D-CNN was utilized to automatically learn a hierarchical spatial representation of the structured web traffic features. CNN models are commonly used in intrusion detection because of their capability to learn discriminative local correlations embedded inside traffic statistics^{9,24}. The Deep Neural Network (DNN) was designed as a fully connected feedforward architecture to capture global nonlinear correlations between traffic features.

Fully connected networks as general baselines can still be useful for intrusion detection when considering their ability to learn complex inter-feature relationships²². In addition, a recurrent neural network based on Gated Recurrent Unit (GRU) was employed to capture the potential sequential dependencies in structured web traffic characteristics. Recurrent models like GRU have achieved strong performance on intrusion detection tasks due to their ability in temporal correlation and contextual relationships capturing with same order of computational complexity compared to standard recurrent neural networks^{10,25}.

To further improve the robustness of classification, a stacking ensemble framework was formed to combine prediction powers coming from the CNN, DNN and GRU models. Ensemble methods have been shown to improve the generalization of multi-class intrusion detection by stacking heterogeneous decision boundaries^{17,22}. The architectural parameters of the implemented models are summarized in **Table 2**, the general structure of the proposed framework is shown in **Figure 1**, and the stacking-based meta-model procedure is presented in **Algorithm (1.1)**.

Table 2. Architectural parameters of the implemented deep learning models.

Model	Main Architecture	Hidden Activation	Output Activation	Drop out	Batch Size	Optimizer
1D-CNN	Conv1D (16,32,48,64) + MaxPooling + Flatten	LeakyReLU	Softmax	0.2	128	AdamW
DNN	Dense (16,32,48,64,32)	LeakyReLU	Softmax	0.2	128	AdamW
GRU-RNN	GRU (16,32,48,64) + Dense (64,32)	LeakyReLU	Softmax	0.2	128	AdamW
Stacking Meta-Model	Dense (128,64) + BatchNormalization	ReLU	Softmax	,0.3 0.4	256	Adam

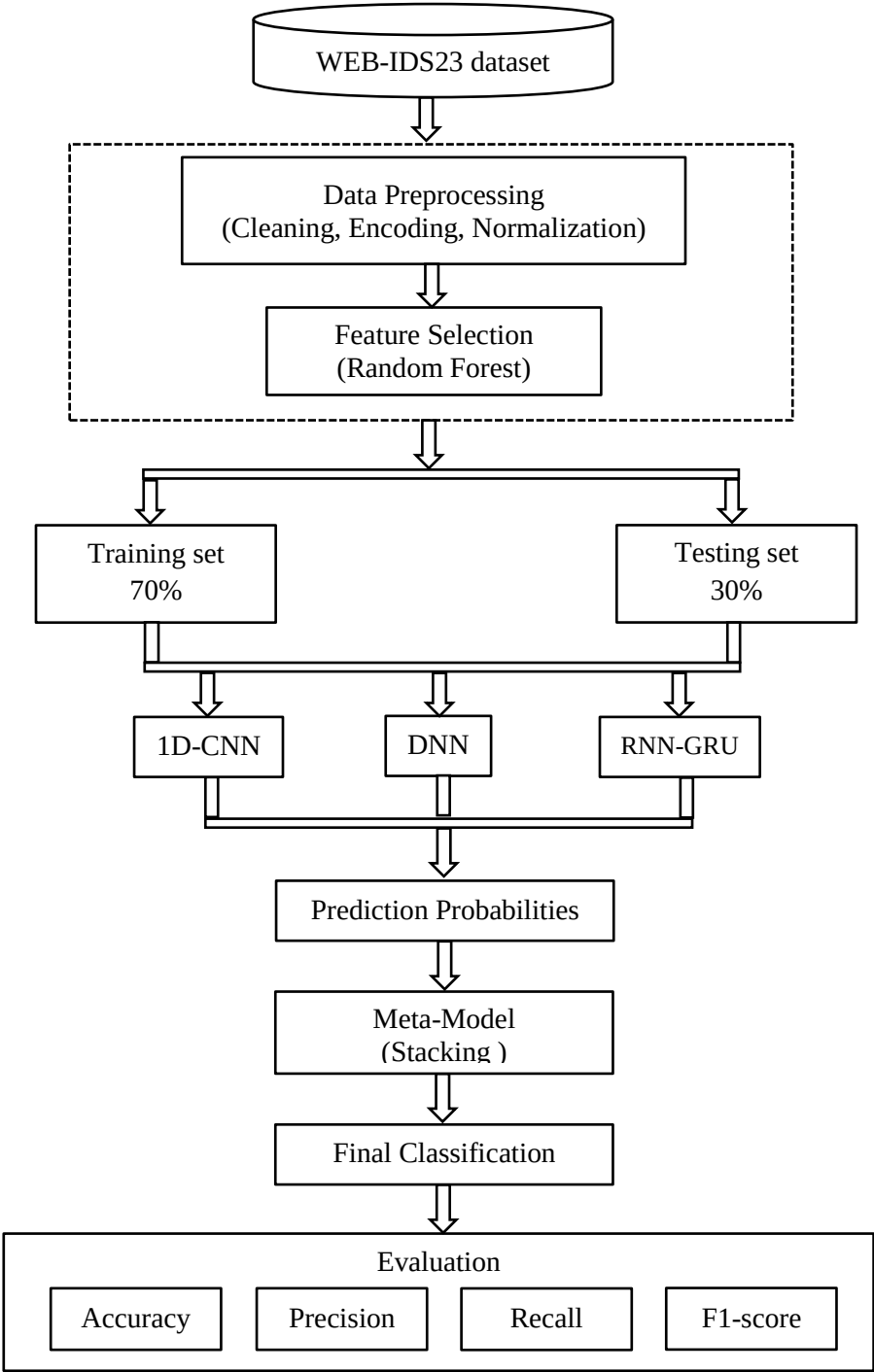


Figure 1. Overview of the proposed deep learning-based web intrusion detection framework using the WEB-IDS23 dataset, including preprocessing, feature selection, model training, and evaluation.

Algorithm (1.1): Stacking-Based Meta Model for Web-Based Attack Classification**Input:** Prediction probabilities generated by CNN, DNN, and GRU-RNN models**Output:** Final attack classification results**Begin**

- Generate prediction probabilities from the trained CNN, DNN, and GRU-RNN models for the training and testing datasets.

- Concatenate the prediction probabilities to construct the training and testing meta-features.

- Allocate 20% of the training meta-features for validation and use the remaining data for training.

- Build the meta-model consisting of:

- 1-Dense (128) + Batch Normalization + Dropout (0.4)

- 2-Dense (64) + Batch Normalization + Dropout (0.3)

- 3-Softmax output layer

- Compile the model using Adam optimizer (learning rate = 0.0008) and sparse categorical cross-entropy loss.

- Configure EarlyStopping with patience = 5 and restore_best_weights = True.

- Train the meta-model for up to 40 epochs with batch size = 256.

- Predict the final attack classes using the testing meta-features.

- Evaluate the model using Accuracy, Precision, Recall, F1-score, Classification Report, and Confusion Matrix.

End

3.Results and Discussion

In this section, the experimental results of 1D-CNN, DNN, GRU-RNN and meta model on the WEB-IDS23 dataset are presented and discussed. All models were trained and evaluated under the same preprocessing, feature selection and evaluation protocol to make a fair comparison.

3.1 Performance Evaluation Metrics and Results

Performance of the models was estimated by four typical evaluation measures: Accuracy, Precision,

Recall, and F1-score. With this the reliability and robustness of the classification under imbalanced data can be measured.

Table 3 shows the overall comparison of CNN, DNN, GRU-RNN, and meta- model deep learning models with respect to detecting accuracy for different categories of web attack.

Table 3. Comparative performance of deep learning models on the WEB-IDS23 dataset.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
1D-CNN	96.43%	96.42%	96.43%	96.32%
DNN	95.89%	95.73%	95.79%	95.39%
GRU-RNN	96.10%	0.9587	95.96%	95.75%
stacking meta-model	96.55%	97.15%	96.55%	96.10%

The 1D-CNN achieved the highest accuracy (96.43%) and F1-score (96.32%), showing its strong ability to capture spatial patterns in web traffic. The GRU-RNN followed and demonstrated effective performance by learning temporal dynamics; the DNN was also consistent in its results at a lower computational cost.

All the models reached to above 95% of accuracy, which proved that deep learning is very effective for intrusion detection. The stacking meta-model contributed to enhance the general classification by combining the predictive capabilities of single base learners. It obtained an accuracy 96.55% and precision 97.15%, indicating more robustness and clearer minority class detection than single models. This result supports our hypothesis that the heterogeneous deep learning model combination with meta-learning has a substantially robust and generalized intrusion detection performance.

3.2 Training and Validation Behavior

The training and validation curves of all models are shown in **Figures 2–5**, where we show their trajectory through training in 100 epochs. All three models showed consistent learning behaviors without observable overfitting, which demonstrated that the selected AdamW

optimizer, and dropout regularization, early stopping techniques were effective. As shown in **Figure 2a**, the 1D-CNN achieved faster convergence with stability than the other networks, while **Figure 2b** shows a gradual decrease in validation loss. **Figures 3** and **4** indicate that the DNN and GRU-RNN showed consistent validation accuracy and balanced loss reduction during training. These findings show that the optimization for all models is balanced, and generalize fairly strong in a class-imbalanced setting.

The stacking meta-model shown in **Figure 5** also demonstrated robust convergence during the training. **Figure 5a** represents the stable validation accuracy during the training epochs, while **Figure 5b** indicates a gradual decrease in validation loss with slight fluctuations, implying that nonlinear relationships among the base model predictions had been effectively learned. The meta-classifier also appeared to provide consistent performance throughout the training epochs in comparison with individual stand-alone models.

In general, the training dynamics show that our framework is able to converge reliably with good generalization performance on all models.

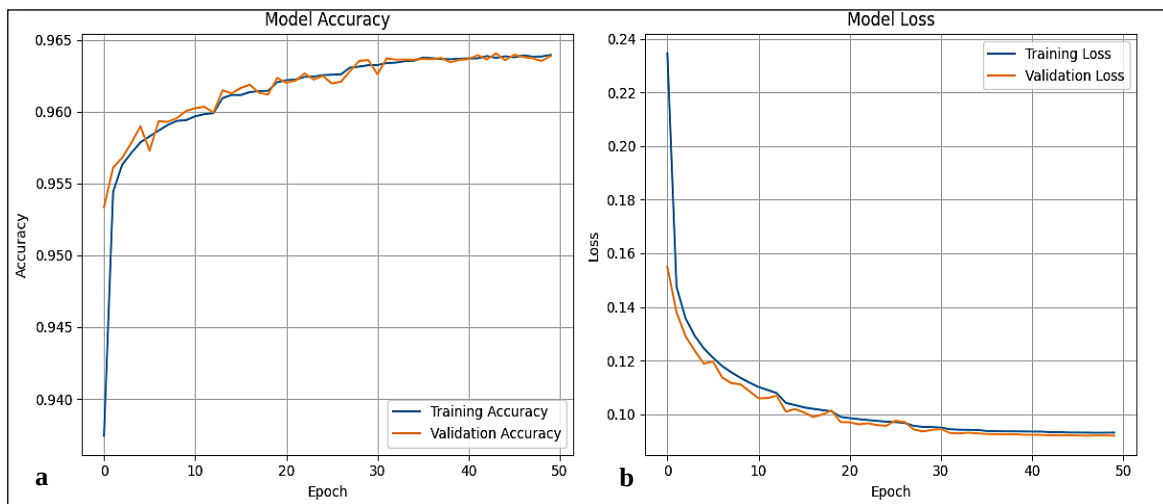


Figure 2. Training and validation performance of the 1D-CNN model: a. accuracy curve, b. loss curve.

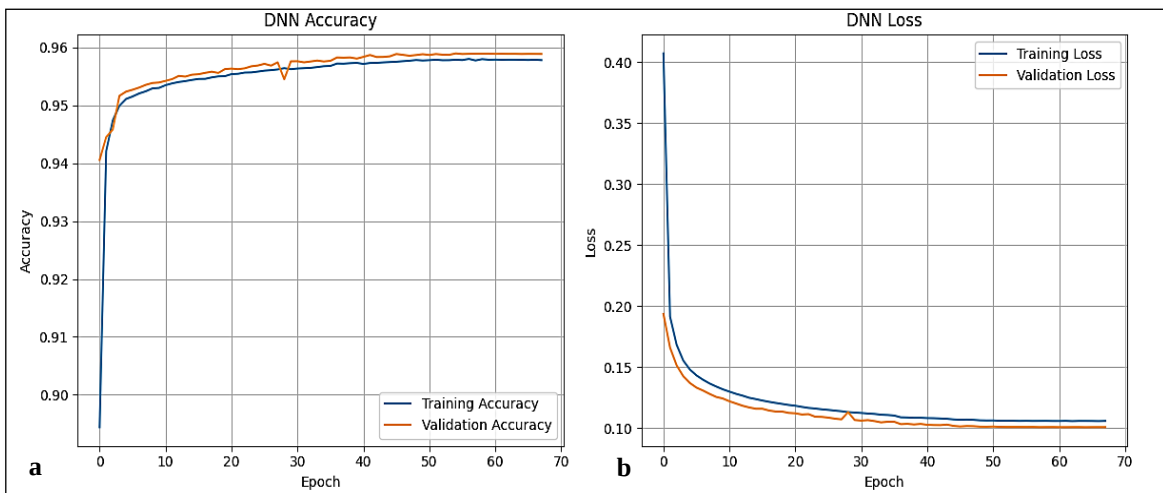


Figure 3. Training and validation performance of the DNN model: a. accuracy curve, b. loss curve.

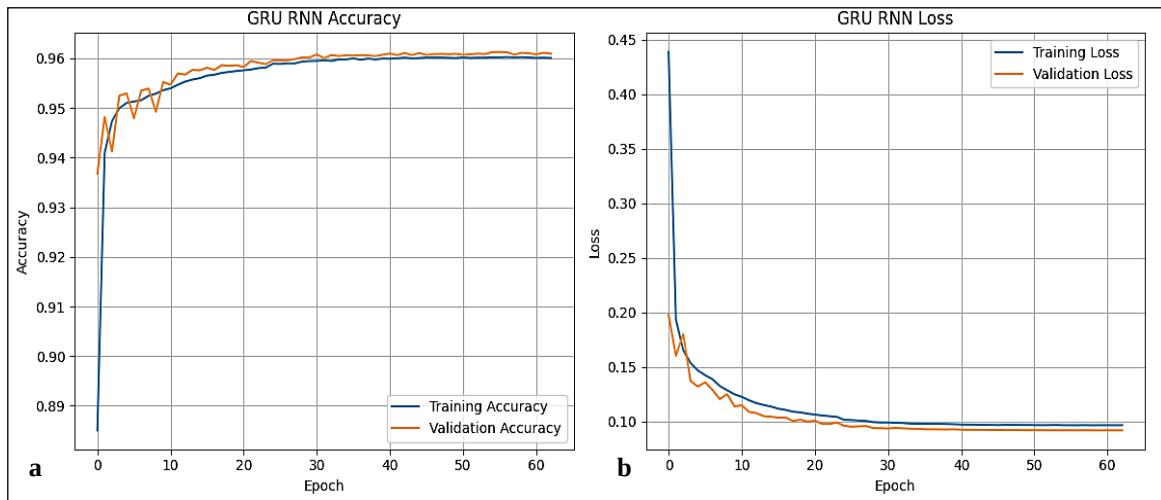


Figure 4. Training and validation performance of the GRU-RNN model: a. accuracy curve, b. loss curve.

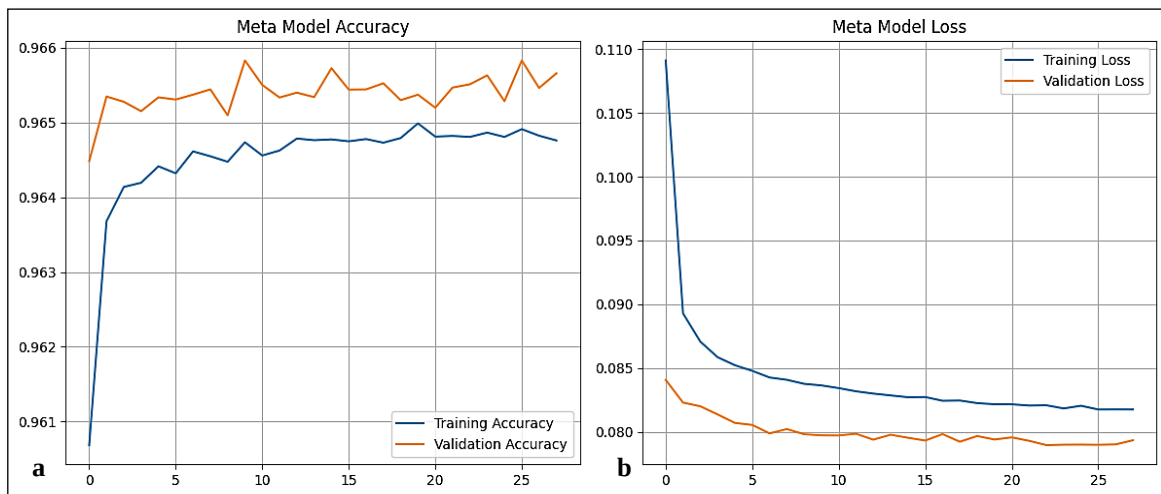


Figure 5. Training and validation performance of the stacking meta-model: a. accuracy curve, b. loss curve.

3.3 Comparative Analysis of Architectures

Using the WEB-IDS23 dataset the experimentation of three models achieved all an accuracy over 95% implies that deep learning-based multi-class web intrusion detection has strong power to prove its robustness. The 1D-CNN achieved the best result in terms of accuracy and F1-score, which was because 1D-CNN is capable of extracting spatial- domain information from web traffic .The GRU-RNN followed, with an accuracy of 96.10%, demonstrating excellent temporal modeling capability to detect sequential attacks like brute-force or reverse shell intrusion. The accuracy of the DNN was 95.89% with stable results and less computational expenses, which can be considered appropriate for lightweight IDSs. Finally, the stacking- based meta-model increased overall accuracy by aggregating probabilities from the three base learners. The fusion model attained an accuracy of 96.55% and improved precision over the individual models, reflecting an improved generalization capability and lower misclassification error rates. Since The individual models already attained high classification accuracy, so the improvement from stacking little. On the other hand, the ensemble model resulted in more robust classification performance and less misclassification across minority attack classes.

The performance comparison of the deep learning models evaluated on the WEB-IDS23 dataset is shown in **Figure 6**.

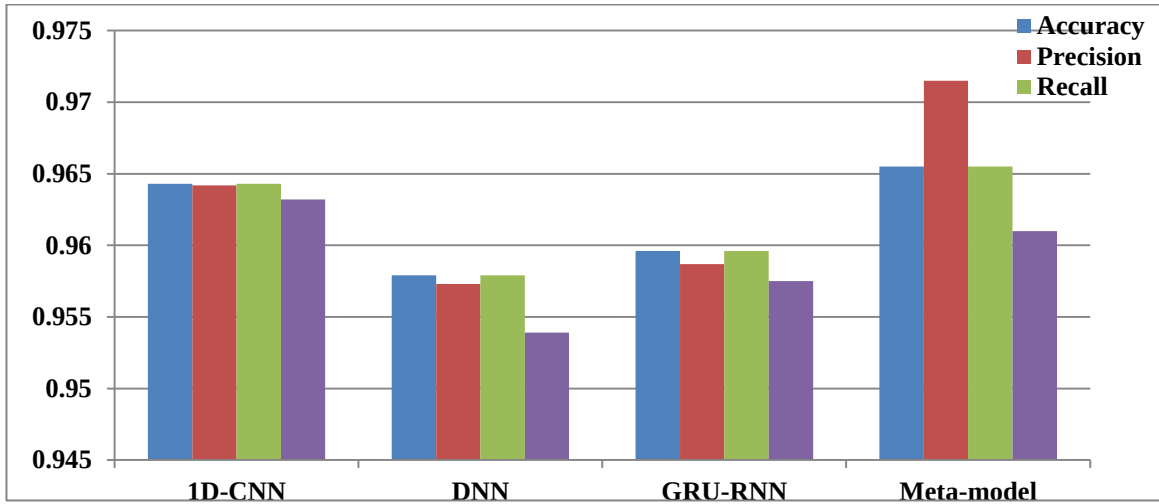


Figure 6. Performance comparison of 1D-CNN, DNN, GRU-RNN and a stacking meta-model in terms of accuracy, precision, recall, and F1-score.

3.4 Per-Class Performance Analysis

The detailed classification reports reveal distinctive performance differences in attack categories, which are mainly due to class imbalance and inter-class feature similarity.

The 1D-CNN model achieved nearly perfect accuracy and sensitivity on the most prominent classes such as benign, bruteforce_http, bruteforce_https, that have significant presence in the dataset. It also had good detection capability on portscan and ssrf_http, indicating that it can effectively catch the discriminative spatial traffic patterns. The per-class performance of GRU-RNN was competitive and had improved recall on some sequential behavior-based attacks, such as portscan and ssh_login_successful; it demonstrated to model the contextual dependencies. Performance of minority classes such as dos_https and smtp_version degraded, but parts of them still performs well. Consequently, the smtp_version class contains only two samples in the test set. As a result, the relevant performance metrics should be interpreted carefully as they are probably statistically unstable. The smtp_version class was preserved for the purpose of retaining the original class distribution of WEB-IDS23 to allow this evaluation to be realistic. The DNN resulted in uniform precision rates on the majority classes, but low recall for a few minority attack types such as xss_http, dos_https and ftp_version. This finding indicates that the dense architectures find it difficult to differentiate fine structure changes when no explicit spatial (or temporal) modeling is enforced. The stacking meta-model delivered a more balanced per-class performance than the individual models. The ensemble mitigated the extreme misclassification of minority categories and increased F1 scores for hard classes including smtp_enum, hostsweep_Pn, and revshell_https. The meta-classifier combined complementary decision patterns through CNN, DNN and GRU outputs for improved robustness and lower variance in performance across attack types. The difference between macro F1-score (0.76) and weighted F1-score (0.96) is primarily due to the class imbalance of WEB-IDS23 dataset. The weighted F1-score is affected by the majority attack classes, as opposed to the macro F1-score, which takes all class behaviours into account evenly around minority attacks too. This suggests that the foundation framework brings greater performance gain on commonly occurring attacks than on less frequent ones.

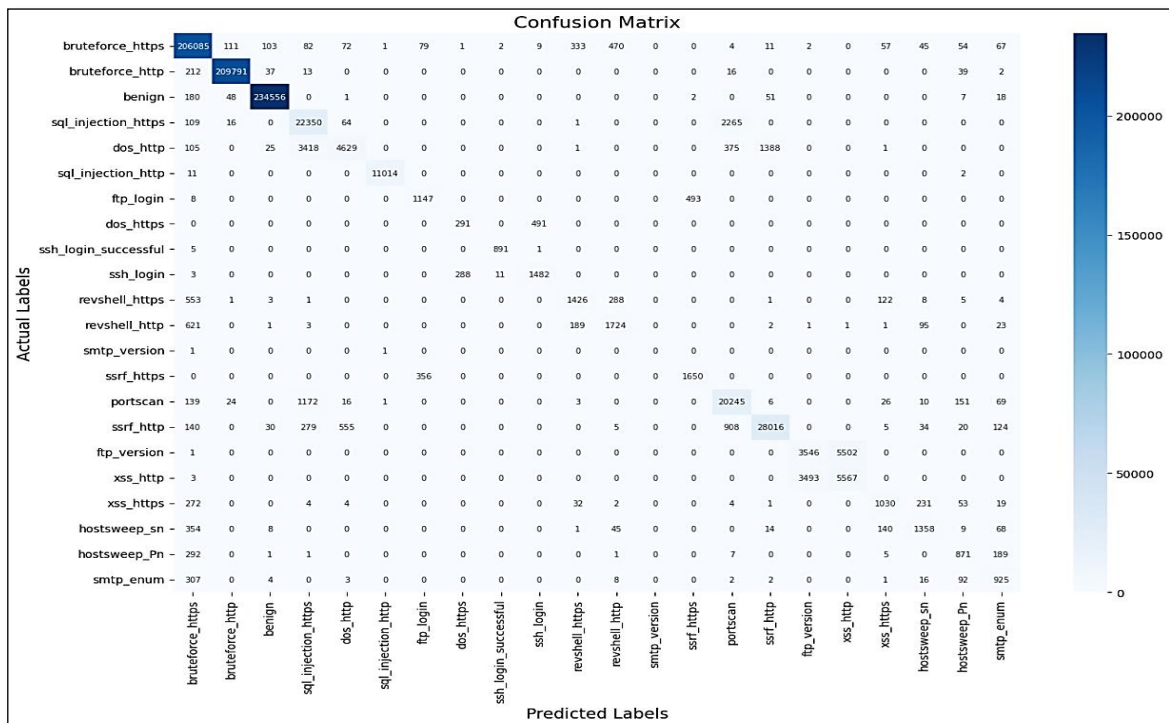
In general, we can conclude that the per-class analysis also shows a negative impact by class imbalance on minority attack detection for both strategies and that the ensemble strategy contributes to higher stability and less disparity of classification results between different web attack categories. **Table 4** shows the summarized classification report of the proposed stacking meta-model for representative attack categories.

Table 4. Classification report of the proposed stacking meta-model

Attack Category	precision	recall	f1-score	support
bruteforce_https	0.98	0.99	0.99	207590
bruteforce_http	1.00	1.00	1.00	210110
benign	1.00	1.00	1.00	234862
sql_injection_https	0.80	0.91	0.85	24805
dos_http	0.84	0.56	0.67	9942
sql_injection_http	1.00	1.00	1.00	11027
ftp_login	0.68	0.76	0.72	1648
accuracy			0.96	786994
macro avg	0.77	0.77	0.76	786994
weighted avg	0.96	0.96	0.96	786994

3.5 Confusion Matrix Interpretation

In addition to overall performance measures, the confusion matrices shown in **Figures 7–10** offer structural information about patterns of inter-class misclassification. As shown in **Figure 7**, The 1D-CNN shows the most concentrated diagonal distribution which reflects good separation between attack families. Misclassifications were predominantly restricted to similar variants such as HTTP/HTTPS siblings, indicative of good spatial discrimination regarding traffic attributes. As shown in **Figure 8**, the DNN model exhibited more widespread diversity among the minority categories and misclassification between structurally overlapping attacks is more likely. This would suggest that we have limitations for simulating subtle inter-class variances without explicit spatial or temporal feature extraction mechanisms. As presented in **Figure 9**, Moderate confusion was observed between behaviorally similar categories, specifically among reverse shells and XSS variants in the GRU-RNN. This pattern indicates that it is sensitive to the sequential similarities of the feature, which may result in blurred differences between attacks with similar temporal dynamics. **Figure 10** represents that the stacking-based meta-model has achieved a cleaner confusion matrix with lesser number of misclassification than individual models. Overall, confusion matrix analysis verifies the effect of architectural diversity and ensemble fusion on enhanced inter-class discrimination performance and robustness against imbalanced traffic conditions.

**Figure 7.** Confusion matrix of the 1D-CNN model.

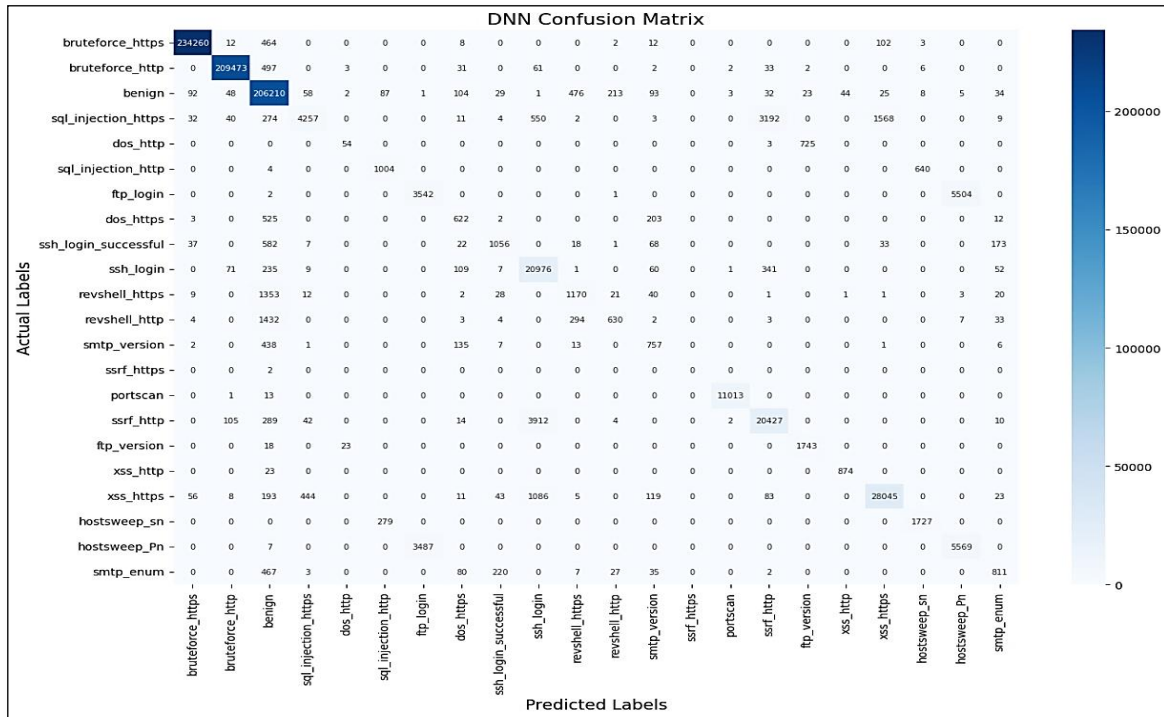


Figure 8. Confusion matrix of the DNN model.

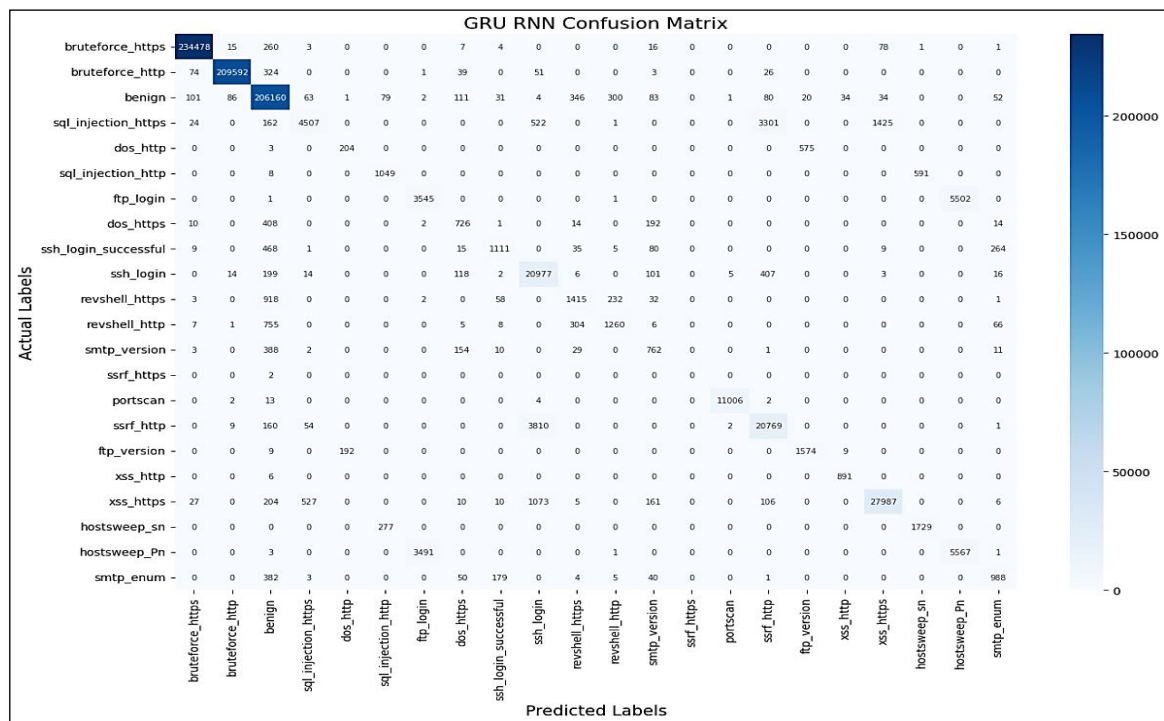


Figure 9. Confusion matrix of the GRU-RNN model

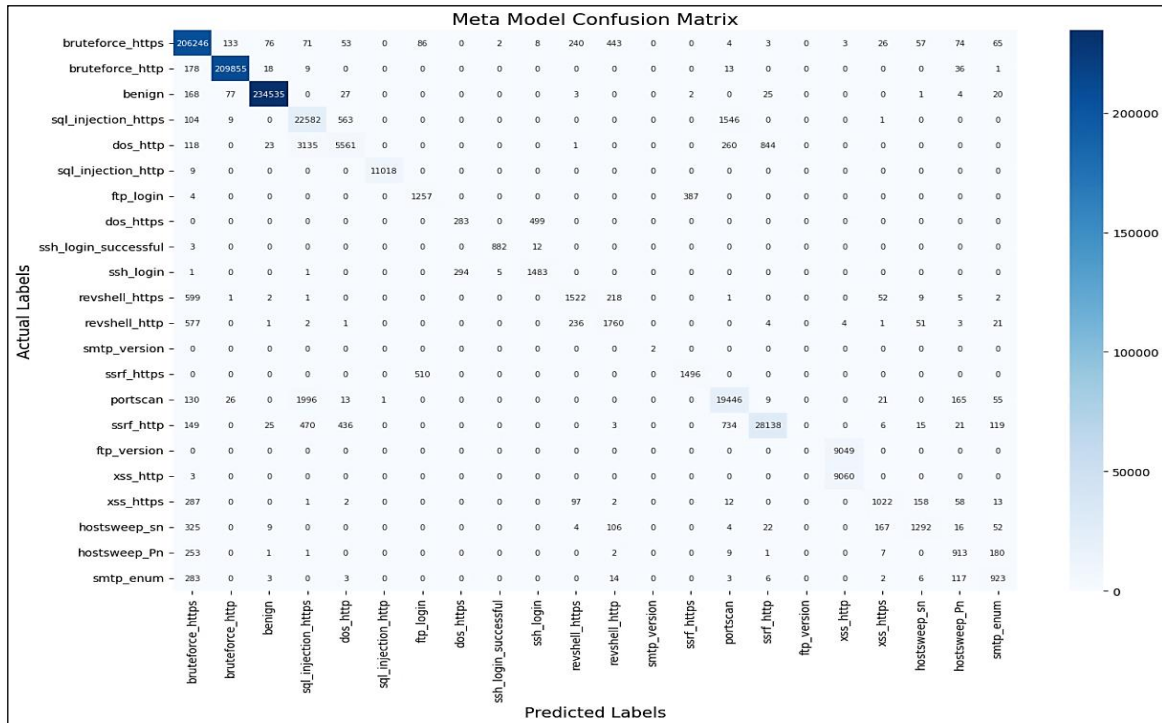


Figure 10. Confusion matrix of the stacking Meta -model

3.6 Experimental Setup

All experiments were conducted using Google Colab Pro with NVIDIA Tesla T4 High-RAM GPU acceleration. The execution environment provided approximately 51 GB of system RAM and 15 GB of GPU memory. The proposed models were implemented using Python 3.10, TensorFlow/Keras, and Scikit-learn libraries. The total training and processing time for all implemented models was approximately seven hours.

4. Conclusion

In this paper, a deep learning model for multi-class web intrusion detection was designed and evaluated using the WEB-IDS23 dataset. The proposed methodology integrates the Random Forest-based feature selection with heterogeneous deep learning models and a stacking meta-learner to improve generalization against class imbalance. The proposed model showed high accuracy for detecting 22 web attack categories. The 1D-CNN outperformed the other isolated models with accuracy 96.43 % and F1-score 96.32 %, while the stacking ensemble method further increased overall accuracy (96.55%) and improved detection stability for minority attack classes. These results verify that learning spatial, temporal, and global features as an ensemble can enhance classification reliability and robustness. However, some limitations such as computational cost and sensitivity to rare types of attacks still exist. In the future, advanced imbalance-handling methods such as SMOTE and cost-sensitive learning, optimization of lightweight models for deployment in real-time, and cross-dataset validation using additional intrusion detection datasets could be explored to improve the generalization capacity. In general, the presented framework provides a scalable, resilient solution for multi-class web intrusions detection in contemporary high-traffic web environments.

Acknowledgment

I would like to express my sincere gratitude to my supervisor, Associate Professor Dr. Amer Abdulmajeed Abdulrahman, for his valuable guidance and continuous support throughout this research. I also extend my appreciation to the Department of Computer Science, Faculty of

Science, University of Baghdad, for their cooperation and assistance.

Conflict of Interest

There are no conflicts of interest to disclose.

Funding

No funding.

References

1. Alghawazi M.; Alghazzawi D.; Alarifi S. Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review. *J Cybersecur Priv.* 2022, 2(4), 764–777. <https://doi.org/10.3390/jcp2040039>.
2. Kumar D.; Pateriya R.K.; Gupta R.K. DDoS Detection Using Deep Learning. *Procedia Comput Sci.* 2023, 2420–2429. <https://doi.org/10.1016/j.procs.2023.01.217>.
3. Tadhani J.R.; Vekariya V.; Sorathiya V. Securing web applications against XSS and SQLi attacks using a novel deep learning approach. *Sci Rep.* 2024, 14(1). <https://doi.org/10.1038/s41598-023-48845-4>.
4. Salam A.; Ullah F.; Amin F. Deep Learning Techniques for Web-Based Attack Detection in Industry 5.0: A Novel Approach. *Technologies.* 2023, 11(4). <https://doi.org/10.3390/technologies11040107>.
5. Kaream D. A. A.; Ali N. H. M. A Robust Base-layer Design for Hierarchical IoT Intrusion Detection Using Hybrid Deep Learning. *Int J Intell Eng Syst.* 2026, 19(5), 606–621. <https://doi.org/10.22266/ijies2026.0531.35>.
6. Dawadi B.R.; Adhikari B.; Srivastava D.K. Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. *Sensors.* 2023, 23(4). <https://doi.org/10.3390/s23042073>.
7. Dai Z.; Por L.Y.; Chen Y.L.. An intrusion detection model to detect zero-day attacks in unseen data using machine learning. *PLoS ONE.* 2024, 19(9). <https://doi.org/10.1371/journal.pone.0308469>.
8. Mohammadpour L.; Ling T.C.; Liew C.S. A Survey of CNN-Based Network Intrusion Detection. *Appl Sci.* 2022, 12(16). <https://doi.org/10.3390/app12168162>.
9. Alabsi B.A.; Anbar M.; Rihan S.D.A. CNN-CNN: Dual Convolutional Neural Network Approach for Feature Selection and Attack Detection on Internet of Things Networks. *Sensors.* 2023, 23(14). <https://doi.org/10.3390/s23146507>.
10. Kasongo S.M. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Comput Commun.* 2023, 199, 113–125. <https://doi.org/10.1016/j.comcom.2022.12.010>.
11. Mahdi S. S.; Abdulrahman A. A. Enhanced CAN Bus Intrusion Detection Using CNN-LSTM-Attention Networks with Time-based Feature Engineering. *Int J Intell Eng Syst.* 2026, 19(1). <https://doi.org/10.22266/ijies2026.0131.34>.
12. Qazi E.U.H.; Faheem M.H.; Zia T. HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System. *Appl. Sci.* 2023, 13(8). <https://doi.org/10.3390/app13084921>.
13. Dhanya K.A.; Vajipayajula S.; Srinivasan K.. Detection of Network Attacks Using Machine Learning and Deep Learning Models. *Procedia Comput Sci.* 2023, 57–66. <https://doi.org/10.1016/j.procs.2022.12.401>.
14. Khan A.A.; Chaudhari O.; Chandra R. A Review of Ensemble Learning and Data Augmentation Models for Class Imbalanced Problems: Combination, Implementation and Evaluation. *Expert Syst Appl.* 2024, 244. <https://doi.org/10.1016/j.eswa.2023.122778>.
15. Alaoui R.L.; Nfaoui E.H. Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review. *Fut Internet.* 2022, 14(4). <https://doi.org/10.3390/fi14040118>.
16. Lanfer E.; Brockmann D.; Aschenbruck N. Technical Report: Generating the WEB-IDS23 Dataset. arXiv preprint arXiv:2502.03909, 2025. <https://doi.org/10.48550/arXiv.2502.03909>.
17. Lazzarini R.; Tianfield H.; Charissis V. A stacking ensemble of deep learning models for IoT intrusion detection. *Knowledge-Based Syst.* 2023, 279. <https://doi.org/10.1016/j.knosys.2023.110941>.
18. Silivery A.K.; Rao Kovvur R.M.; Solleti R. A model for multi-attack classification to improve intrusion detection performance using deep learning approaches. *Meas Sens.* 2023, 30. <https://doi.org/10.1016/j.measen.2023.100924>.

19. Zhang Y.; Muniyandi R.C.; Qamar F. A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance. Appl. Sci. 2025, 15(3). <https://doi.org/10.3390/app15031552>.
20. Umar M.A.; Chen Z.; Shuaib K.. Effects of feature selection and normalization on network intrusion detection. Data Sci Manag. 2025, 8(1), 23–39. <https://doi.org/10.1016/j.dsm.2024.08.001>.
21. Hashmi A.; Barukab O.M.; Osman A.H. A hybrid feature weighted attention based deep learning approach for an intrusion detection system using the random forest algorithm. PLoS ONE. 2024, 19(5). <https://doi.org/10.1371/journal.pone.0302294>.
22. Wang Y.; Liu Z.; Zheng W. A Combined Multi-Classification Network Intrusion Detection System Based on Feature Selection and Neural Network Improvement. Appl Sci. 2023, 13(14). <https://doi.org/10.3390/app13148307>.
23. Alghofaili L.N.; Ibrahim D.M. Web-Based Attacks Detection Using Deep Learning Techniques: A Comprehensive Review. Indones. J Electr Eng Comput Sci. 2025, 39(1), 466. <https://doi.org/10.11591/ijeecs.v39.i1.pp466-484>.
24. Kamil W. F.; Mohammed I. J. Adapted CNN-SMOTE-BGMM Deep Learning Framework for Network Intrusion Detection using Unbalanced Dataset. Iraqi J Sci. 2023, 64(9), 4846–4864. <https://doi.org/10.24996/ijs.2023.64.9.43>.
25. Hnamte V.; Hussain J. DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System. Telemat Inform .Rep. 2023, 10. <https://doi.org/10.1016/j.teler.2023.100053>.