International Journal of Applied Mathematics

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

OPTIMIZING CONVOLUTIONAL NEURAL NETWORKS FOR WEB SECURITY WITH A HYPERPARAMETER TUNING TO ENHANCE MODEL PERFORMANCE

[1]Ravinder Singh, [2]Dr. Dipak Raghunath Patil, [3]Dr. Mukesh Kumar Gupta, [4]Dr. Sarang M. Patil

[1]Department of Computer Engineering, Suresh Gyan Vihar University, Jaipur, India.
[2]Department of Computer Engineering, Amrutvahini COE, Sangamner, India,
[3]Department of Electrical Engineering, Suresh Gyan Vihar University, Jaipur, India,
[4]Department of Computer Science & Engineering, Amity School of Engineering & Technology, Amity University Mumbai, India,
singhravindersingh@gmail.com, Patildipak87@gmail.com, mkgupta72@gmail.com, smpatil@mum.amity.edu

Abstract

Web applications are prime targets for cyber-attacks due to their network accessibility and inherent vulnerabilities. Traditional web security mechanisms such as rule-based intrusion detection systems and conventional machine learning models are mostly work on predefined rules and extensive labelled datasets which makes them less effective against evolving threats. This paper explores an optimized hyperparameter tuning approach to enhance the performance and efficiency of convolutional neural networks (CNNs) for web security. Our contributions are threefold. First, we trained the CNN model on pre-processed of unstructured and imbalanced datasets collected from open repositories. Second, hyperparameters of CNN are optimized on different constant batch sizes like 32 and 16 to improve model accuracy of CNN models. Third, the proposed models are trained again on the optimized hyperparameters values for better results of accuracy. This research highlights the potential of optimized CNN model in revolutionizing web security to provide an intelligent and autonomous solution to counteract cyber-attacks.

Keywords: Web Attacks, CNN Model, Vulnerabilities, Hyperparameters, Batch Size, Cybersecurity

1. Introduction

With the rapid expansion of websites and web applications, the risk of web vulnerabilities is continuously increasing. A study reported that 90% of web applications are vulnerable with 68% at risk of sensitive data breaches [1] and 8% of network intrusions of unauthorized access to web servers. The widespread and diverse use of the internet makes the web applications a prime target for cyber-attacks for criminals. Protecting these applications from cyber-attacks

Received: July 22, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

such as cross-site scripting (XSS) and SQL injection etc has now become a crucial aspect of cybersecurity. Traditional web security solutions that include rule-based Web Application Firewalls (WAFs) and signature-based intrusion detection systems are struggling to keep up with evolving attack [2]. Machine learning (ML) approaches have improved attack detection capabilities but they are also often requiring a large number of labeled datasets, which makes them more expensive and impractical for real-world applications [3]. To address these challenges, deep learning-based techniques like CNNs, LSTMs etc have emerged as powerful tools for detecting web vulnerabilities [4]. However, optimizing these models for efficiency and accuracy performance are also remain a significant challenge.

The unaddressed web vulnerabilities led to severe consequences of data breaches, unauthorized access, and service disruptions. Various security methodologies have been proposed to counter these vulnerabilities and each played a crucial role at different stages of the web application security. These countermeasures were implemented at either the client-side or server-side to strengthen the overall security of web applications [5]. Web security approaches primarily include secure programming, static analysis, dynamic analysis, and black-box fuzzing, each offered unique advantages and limitations. Secure programming focused on best coding practices such as input validation, query parametrization, and adherence to security standards. However, secure programming alone is insufficient due to the complexity and evolving nature of web applications.

Static analysis inspects the application's source code without execution to detect vulnerabilities like SQL injection and Cross-Site Scripting (XSS), but it often produces false positives and struggles with large codebases [6]. Dynamic analysis involves executing the application and identifying security violations through methods such as fuzzing and taint analysis, providing accurate results but with limited code coverage. Lastly, black-box fuzzing tests applications without prior knowledge of their internal workings, using randomized inputs to uncover vulnerabilities, particularly XSS, injection attacks and distributed denial-of-service (DDoS) attacks [7]. These threats compromise sensitive data, disrupt services, and cause significant financial and reputational damage. Traditional machine learning techniques have been widely employed for attack detection, but their ability to extract deep features extraction is often limited due to shallow architectures and manual feature engineering. In contrast, deep learning, particularly CNs, has demonstrated remarkable success in web security applications by automatically learning features from network traffic data [8]. CNNs have shown superior performance in cyberattack detection, intrusion detection, and malware classification. However, optimizing CNN architectures for web security requires careful tuning of hyperparameters such as learning rate, batch size, activation functions, and convolutional filter sizes to enhance detection accuracy and computational efficiency [9].

In recent years, researchers have explored various deep learning models to strengthen cybersecurity. Notable works include literature reviews on deep learning-based attack detection [10], provided an extensive overview of deep learning advancements in network security [11],

Received: July 22, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

examined deep learning techniques for intrusion and malware detection. analyzed cyber threats targeting IoT networks was analyzed [12], while network security datasets were categorized and evaluated deep learning models on benchmark datasets like CSE-CIC-IDS2018 and Bot-IoT [13]. Despite these advancements, optimizing CNN architectures through hyperparameter tuning remains an open research area, with the potential to significantly improve detection accuracy and reduce false positives. Hyperparameter tuning plays a critical role in enhancing the effectiveness of deep learning models by optimizing key parameters such as learning rates, activation functions, batch sizes, and network architectures [14,15]. By systematically fine-tuning these parameters, the proposed approach aims to achieve higher detection accuracy, reduced false positives, and improved computational efficiency. The contributions of this research include:

- Developing an optimized CNN for web attack detection with enhanced performance and efficiency.
- Implementing hyperparameter tuning techniques to refine model architecture and improve threat detection accuracy.
- Evaluating detection performance on optimized hyperparameters to ensure robustness and scalability of the proposed model.

By optimizing CNNs with hyperparameter tuning, this research aims to provide an advanced and efficient web security framework capable of detecting and mitigating emerging cyber threats with greater accuracy and reliability. To the best of our knowledge, this study is one of the first to systematically explore the optimization of CNNs for web security using hyperparameter tuning techniques to enhance both detection accuracy and computational efficiency. While deep learning has been extensively applied to cybersecurity, most existing surveys have not specifically addressed the role of CNN hyperparameter optimization in web attack detection.

2. Literature Survey

Several related studies have investigated various aspects of web security. A comprehensive survey on detecting and mitigating web vulnerabilities was conducted [16-19], providing an in-depth analysis of web security threats and traditional countermeasures. However, their review primarily focused on traditional ML based attack detection approaches, without delving into the optimization of deep learning models like CNNs. Other surveys have explored the applications of ML and DL in broader cybersecurity contexts. However, they did not emphasize web application security specifically, nor did they analyze hyperparameter tuning strategies for improving model performance. Studies [20, 21] focused on web vulnerability classification and countermeasures, but they did not investigate deep learning-based solutions or follow a systematic literature review (SLR) protocol for structured evaluation.

In contrast, study conducted an SLR on web services attacks and security, presenting a structured overview of various cyber threats [22, 23]. Studies provided a more recent survey

Received: July 22, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

highlighting machine learning and deep learning-based techniques for detecting Cross-Site Scripting (XSS) attacks, but without focusing on CNN optimization or hyperparameter tuning techniques. The integration of deep learning in cybersecurity has significantly improved the detection and mitigation of web-based attacks. Several studies have explored the application of deep learning models for enhancing cybersecurity, particularly in web attack detection [24, 25]. A comprehensive review of deep learning applications in cybersecurity has examined various deep learning models, techniques, and their effectiveness in detecting and mitigating cyber threats [26. The study provides insights into the advantages of deep learning over traditional machine learning methods but does not focus on CNN optimization strategies or hyperparameter tuning for enhanced web security [27].

Another relevant survey discussed the evolution of end-to-end deep learning models in cybersecurity, with a particular emphasis on web attack detection. This work underscores the potential of deep learning architectures in providing holistic solutions but does not analyze performance enhancements achieved through hyperparameter tuning in CNNs. The study [28] investigates the security challenges associated with adversarial attacks on deep learning-based cybersecurity systems. It explores various defense mechanisms but does not focus on optimizing CNNs for robust and efficient web attack detection.

The survey examined the feasibility of real-time attack detection using deep learning [29]. While it highlights the benefits and challenges of implementing deep learning models in real-time scenarios, it does not delve into CNN architecture optimization or hyperparameter tuning techniques that could improve both accuracy and efficiency in web security applications. Despite these significant contributions, existing literature lacks a dedicated focus on optimizing CNNs for web security through hyperparameter tuning. CNNs have shown great potential in detecting complex web threats, but their performance can be significantly influenced by hyperparameters such as learning rate, batch size, filter size, activation functions, and optimization algorithms. Efficient tuning of these parameters can lead to improved detection accuracy, reduced computational overhead, and better generalization across various attack types.

3. Research Methodology

To achieve the objectives outlined in this study, we adopt a structured research methodology that consists of six key phases: Data Collection, Data Preprocessing, Data Transformation, Model Design, Hyperparameter Tuning, Training & Evaluation. Each phase ensures the development of an optimized CNN-based web security framework that effectively detects attacks while improving performance and efficiency.

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

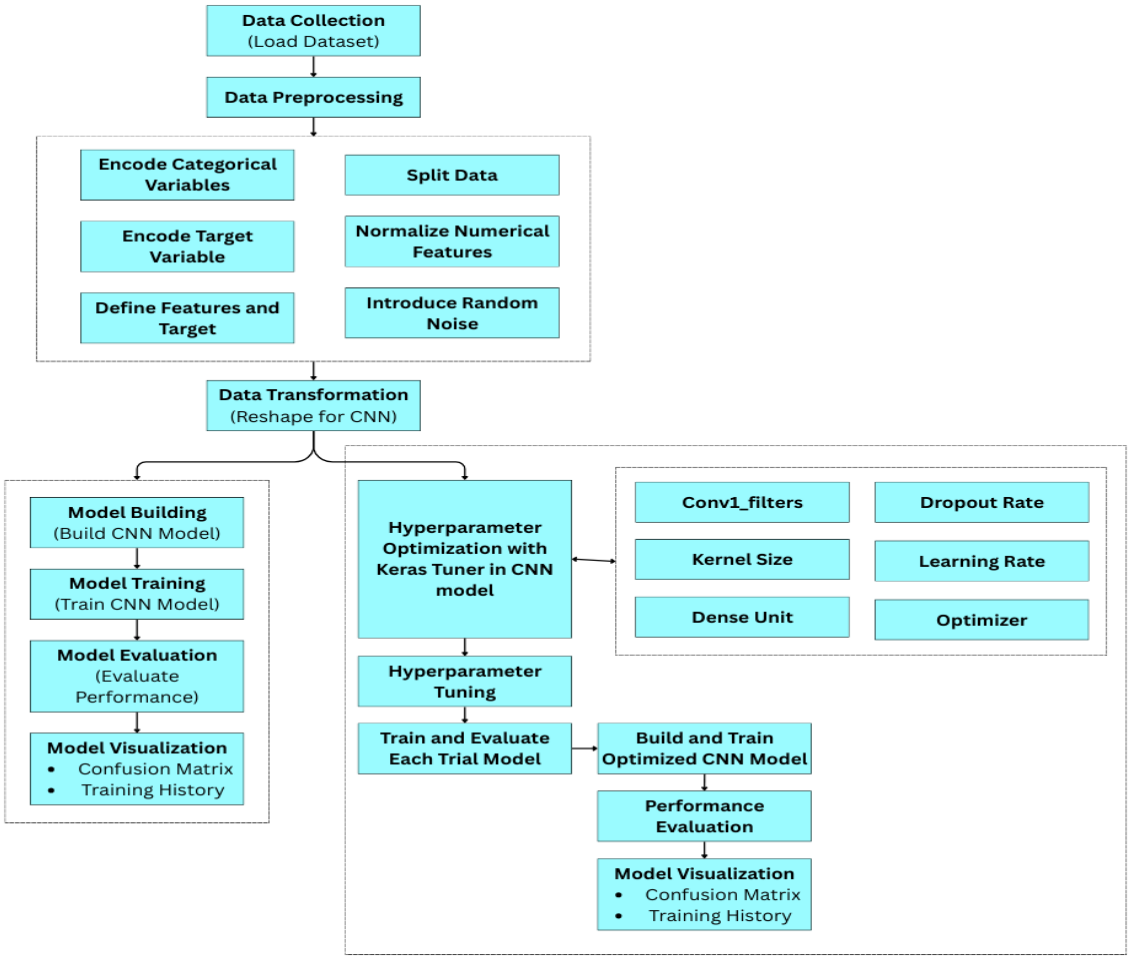

Figure 1: Optimized CNN-based web security framework

We collect diverse web attack datasets from publicly available repositories, including CICIDS 2017, OWASP Web Attack Dataset, and other real-world web logs. A useful dataset is prepared from the collected information by extracting relevant data for further pre-processing. The pre-processing steps involve removing noisy, incomplete, and redundant data using data imputation and filtering techniques. Additionally, categorical web request data is converted into numerical format using One-Hot Encoding, Word Embeddings (Word2Vec), and TF-IDF vectorization. To ensure uniform CNN model performance, Min-Max Scaling is applied for feature normalization. Data imbalance issues are addressed using SMOTE (Synthetic Minority Over-sampling Technique) to prevent biased learning. Finally, the dataset is reshaped to make it compatible with the CNN model, which is designed for web attack detection and deep learning-based training.

To detect malicious web requests, we design a CNN-based deep learning model with a structured architecture. The Input Layer processes transformed web request features, while Convolutional Layers extract spatial and sequential patterns from HTTP request sequences. Pooling Layers reduce dimensionality while preserving essential attack patterns. The Fully

Received: July 22, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Connected Layers classify web requests as normal or malicious, and the Output Layer employs a Softmax/Sigmoid Activation Function to generate attack probability scores. This structured CNN model effectively identifies cyber threats by analyzing complex patterns in web request data.

To optimize model performance, we conduct hyperparameter tuning using Bayesian Optimization and Grid Search. The key hyperparameters considered for tuning include learning rate, batch size, kernel size, dropout rate, number of dense layers, number of filters, and optimizer selection. We employ K-Fold Cross-Validation (K=5) to evaluate the model across multiple trials with different hyperparameter combinations, ensuring robust and optimal tuning. This approach enhances model generalization and improves its capability to detect diverse web attacks efficiently.

In the training process, we use 80% of the dataset for model training and 20% for validation on unseen data. Early stopping is implemented to prevent overfitting, ensuring that the model does not learn unnecessary patterns. Additionally, we leverage GPU acceleration (TensorFlow with CUDA on NVIDIA GPUs) to speed up the training process and handle large datasets efficiently. To measure model efficiency and accuracy, we evaluate it using Precision, Recall, and F1-Score, which provide insights into classification quality. By leveraging an optimized CNN model with advanced hyperparameter tuning techniques, our approach ensures high accuracy, fast inference, and adaptability to emerging web threats. This makes it a practical and effective solution for modern web security challenges.

4. Results and Discussion

The proposed models of deep learning like CNN is trained as per the above methodology given in figure 1. The experiment results obtained from the proposed model with and without optimizing hyperparameters values are hereby are analysed and discussed accordingly.

In the first stage, the CNN model is trained on the pre-processed dataset on epochs 50 and batch size 32, and 16 without optimizing the hyperparameters. The performance metrics of of proposed model is given in the table 1.

Table 1: Model Performance Metrics on batch size 32

Model	Batch Size	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Support
CNN	**32**	**96.45**	**96.48**	**96.45**	**96.41**	**141**
CNN	16	91.49	91.74	91.49	91.08	141

The CNN model for the batch size 32 is giving superior performance compared the model is trained for the batch size 16 on the same epochs as shown in table 1. The matrices like accuracy of 96.45% of CNN is obtained on the 141 support, and similar other parameters of metrics.

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Hyperparameters Tuning

In this paper, hyperparameters are tuned for the better accuracy of the proposed models. Here the model is trained on five trials of hyperparameters combinations for the both batch of 32 and 16. Each model is run for the 50 epochs.

The hyperparameters combinations of five trails for the CNN model are given in the table 1 for of batch size 32. The outperform accuracy 97.16% is obtained on trial 2 hyperparameters. Simmialrly the superior accuracy 94.33% for the same model on batch size 16 is obtained on trial 3 and 4 as given in the table 2.

Table 2: Hyperparameters and accuracy of CNN model (batch size: 32)

Trial	**Hyperparameters**						**Accuracy (%)**
	Conv1 Filters	**Kernal Size**	**Dense Layers**	**Learning Rate**	**Dropout Rate**	**Optimizer**	
1	96	2x2	64	0.02258	0.2	Adam	95.74
2	**64**	**3x3**	**64**	**0.01351**	**0.3**	**RMSprop**	**97.16**
3	32	2x2	128	0.00333	0.2	RMSprop	96.45
4	32	3x3	128	0.00307	0.2	RMSprop	96.45
5	32	2x2	96	0.00217	0.2	SGD	75.89

Table 3: Hyperparameters and accuracy of CNN model (batch size: 16)

Trial	**Hyperparameters**						Accuracy (%)
	Conv1 Filters	**Kernal Size**	**Dense Layers**	**Learning Rate**	**Dropout Rate**	**Optimizer**	
1	96	2x2	64	0.02258	0.2	Adam	93.62
2	64	3x3	64	0.01351	0.3	RMSprop	90.78
3	**32**	**2x2**	**128**	**0.00333**	**0.2**	**RMSprop**	**94.33**
4	**32**	**3x3**	**128**	**0.00307**	**0.2**	**RMSprop**	**94.33**
5	32	2x2	96	0.00217	0.2	SGD	75.89

Table 4: Hyperparameter tuning for CNN model

Hyperparameter	**Tuning Strategy**	**Optimized Value**
Learning Rate	Scheduling (0.0001–1.0	0.01351
Dense Layers	Neural architecture (32–128)	64
Batch Size	Grid Search {16, 32}	32
Number of Filters	Grid search (32–128)	64
Kernel Size	Experimentation (2×2, 3×3, 5×5)	3×3

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Dropout Rate	Regularization (0.2–0.5)	0.3
Optimizer	Adam, RMSprop, SGD	RMSprop

Table 5: Impact on accuracy with optimized values of hyperparameters

Optimized Values of Hyperparameters							**Impact on accuracy (%)**
Conv1 Filters	**Kernal Size**	**Dense Layers**	**Learning Rate**	**Dropout Rate**	**Batch Size**	**Optimizer**	
64	3x3	64	0.01351	0.3		RMSprop	+0.71
32	2x2	128	0.00333	0.2	16	RMSprop	+2.84
32	3x3	128	0.00307	0.2	16	RMSprop	+2.84

The optimized value of hyperparameters between the batch size 32 and 16 is illustrated in the table 4 for the better accuracy of the proposed model. It is observed from the experiments that the impact on accuracy (+2.84) is more prominent if the CNN model is trained for the batch size 16 and same epochs 50 with optimized values of hyperparameters as illustrated in table 5.

The confusion matrix of the CNN model for batch size 32 without hyperparameter tuning is illustrated in Figure 2. This confusion matrix represents the relationship between the actual and predicted values, where 65 indicates a high matching score, while 4 represents the critical score. The training and validation accuracy along with the loss of the CNN model for batch size 32 without hyperparameter tuning is shown in Figure 3. The validation accuracy reaches 96.45% after 50 epochs, while the loss is 0.1 for the same number of epochs.

Similarly, the confusion matrix for batch size 16 without hyperparameter tuning is illustrated in Figure 4, where the highest match score is 63. The training and validation accuracy along with the loss curve is shown in Figure 5 for batch size 16 without hyperparameter tuning. The accuracy for batch size 16 is 91.49%, which is 4.96% lower compared to the CNN model trained with batch size 32 without hyperparameter tuning.

The results of the CNN model without hyperparameter tuning indicate that the model performs better with a batch size of 32 compared to batch size 16. Hence, the model is now trained again using optimized hyperparameter values to further enhance its performance.

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

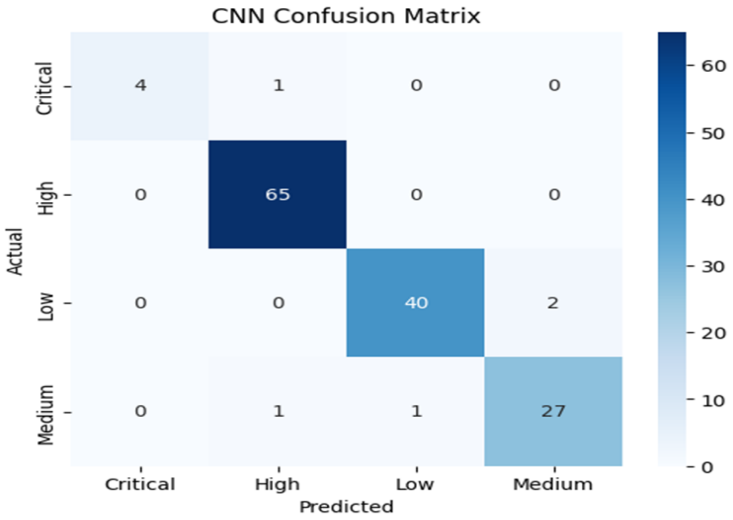

Figure 2: Confusion matrix without hyperparameters tuning (batch size: 32)

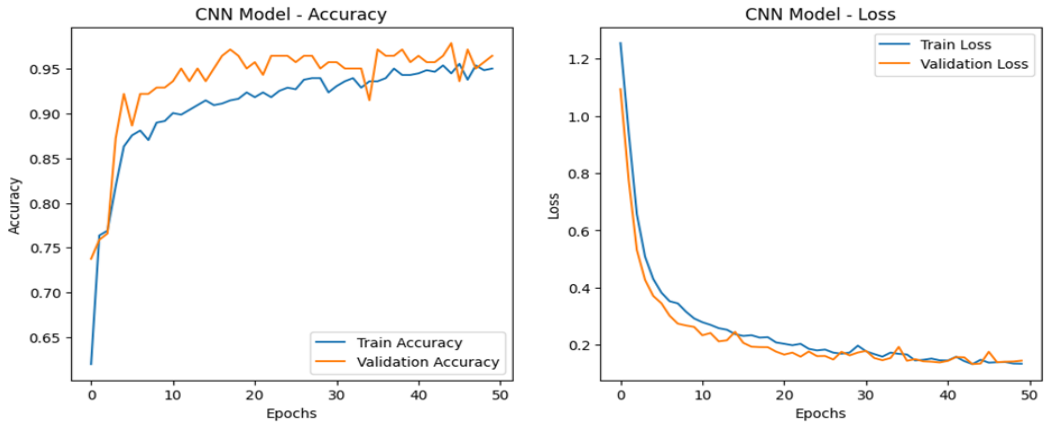

Figure 3: Accuracy and loss graph without hyperparameters tuning (batch size: 32)

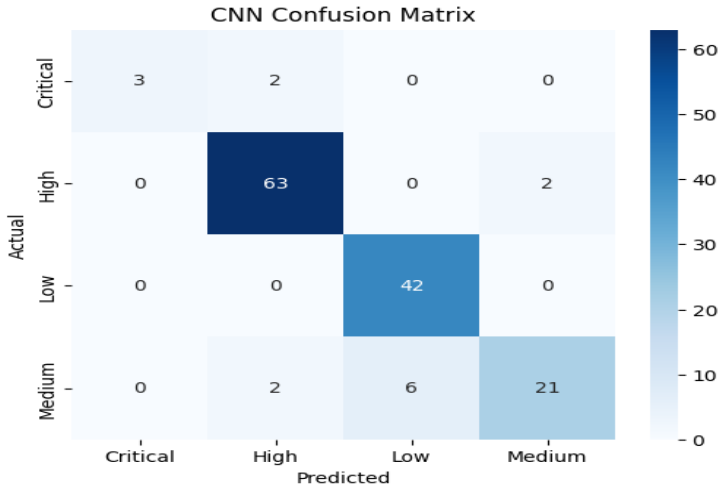

Figure 4: Confusion matrix without hyperparameters tuning (batch size: 16)

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

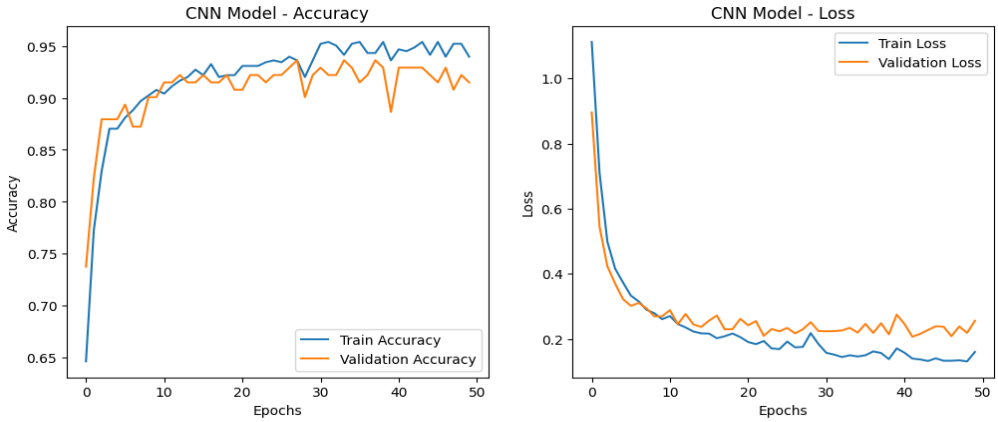

Figure 5: Accuracy and loss graph without hyperparameters tuning (batch size: 16)

The confusion matrix of the CNN model for batch size 32 with hyperparameter tuning is illustrated in Figure 6, where 64 indicates a high matching score, while 3 represents the critical score. The training and validation accuracy, along with the loss of the CNN model for batch size 32 with hyperparameter tuning, is shown in Figure 7. The validation accuracy reaches 97.16% after 50 epochs in trial 2, using optimized hyperparameter values such as 64 filters, kernel size of 3×3, 64 dense layers, learning rate of 0.01351, dropout rate of 0.3, and the RMSprop optimizer, while the loss is 0.07 for the same number of epochs. The accuracy increased by 0.71% after hyperparameter tuning.

Similarly, the confusion matrix for batch size 16 with hyperparameter tuning is illustrated in Figures 8 and 10, where the highest match score is 64 and 63, respectively. The training and validation accuracy, along with the loss curve, is shown in Figures 9 and 11 for batch size 16 with hyperparameter tuning. The accuracy obtained for batch size 16 using the optimized hyperparameters in trials 3 and 4 is 94.33%, which is 2.84% higher compared to the CNN model trained with batch size 16 without hyperparameter tuning. The results of the CNN model with hyperparameter tuning indicate that the model performs better with a batch size of 32 compared to batch size 16.

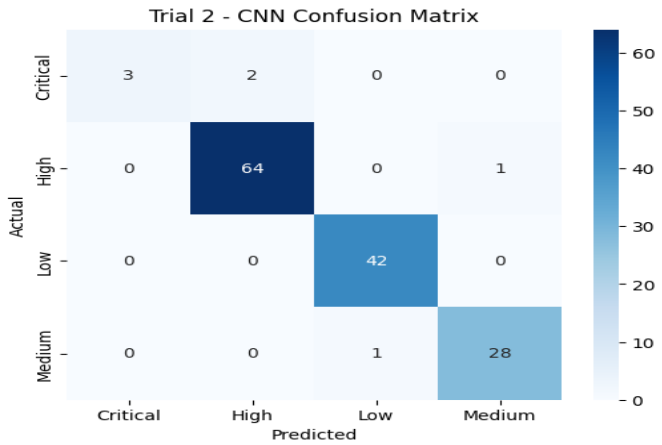

Figure 6: Confusion matrix with hyperparameters tuning (batch size: 32)

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

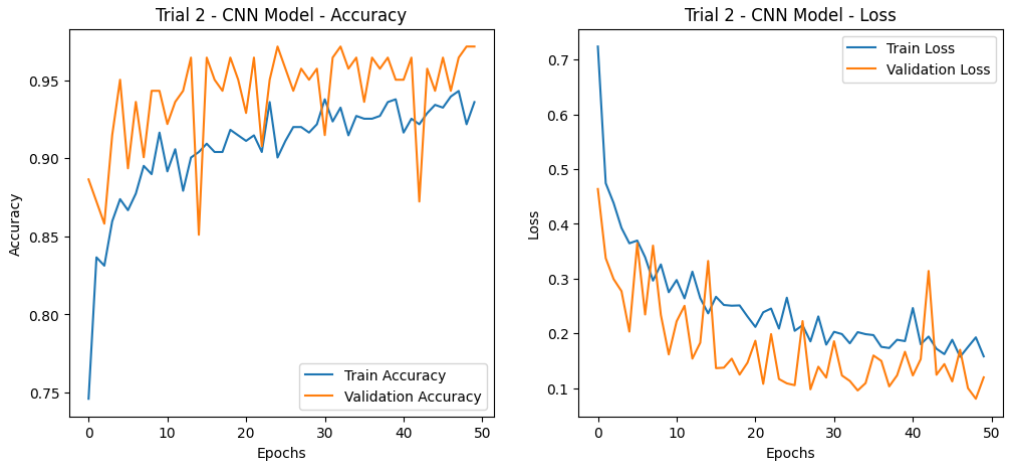

Figure 7: Accuracy and loss graph with hyperparameters tuning (batch size: 32)

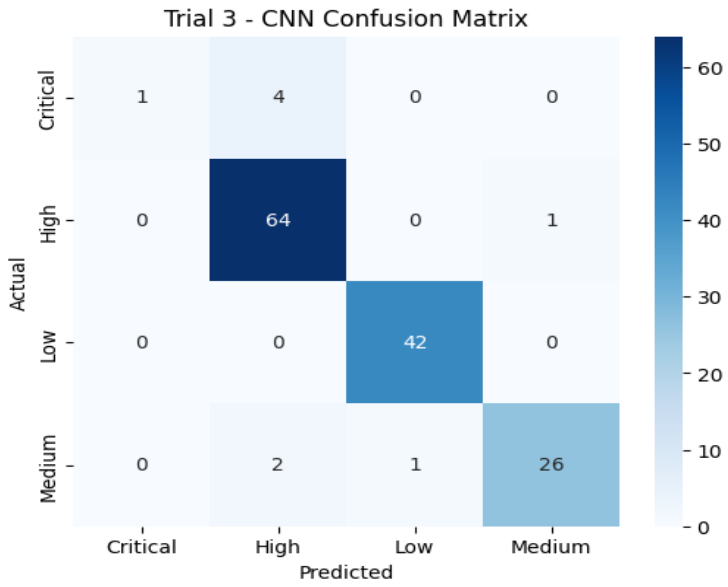

Figure 8: Confusion matrix with hyperparameters tuning (batch size: 16)

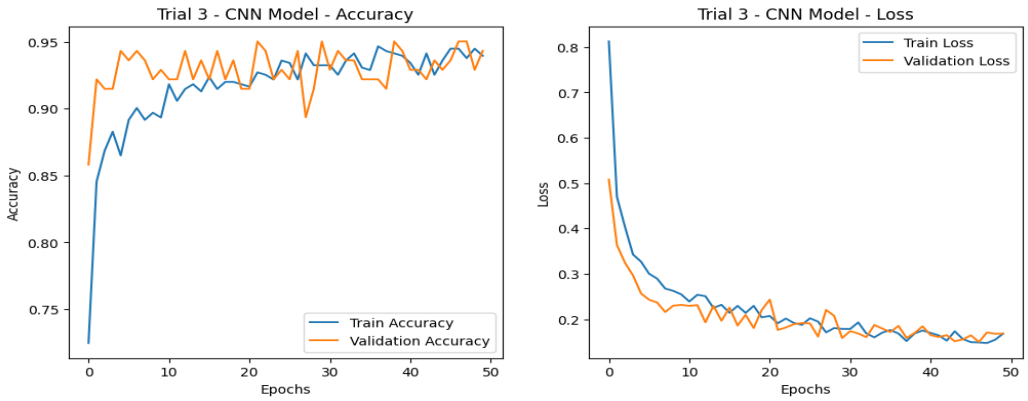

Figure 9: Accuracy and loss graph with hyperparameters tuning (batch size: 16)

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

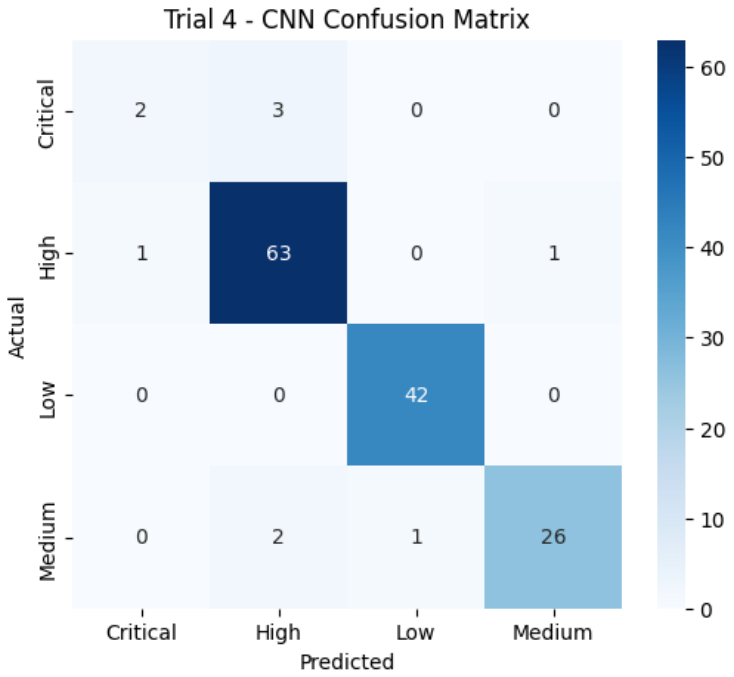

Figure 10: Confusion matrix with hyperparameters tuning (batch size: 16)

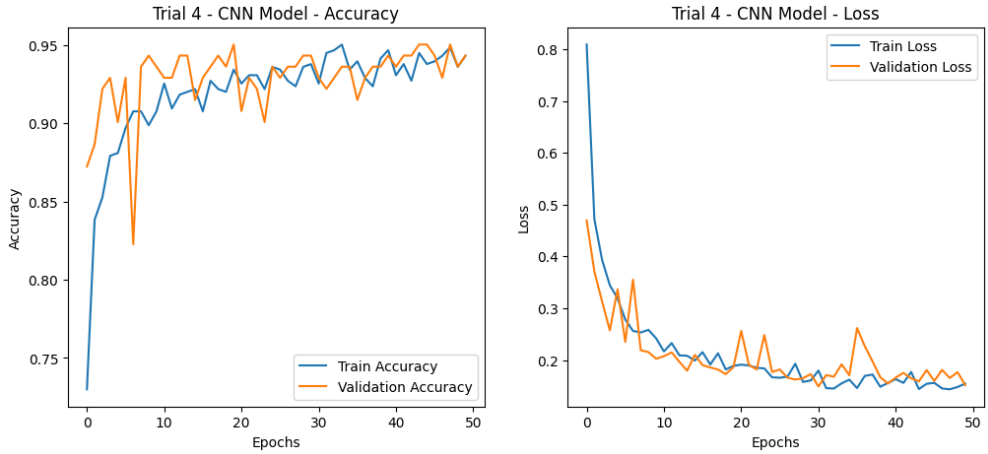

Figure 11: Accuracy and loss graph with hyperparameters tuning (batch size: 16)

Conclusion

Web applications are highly vulnerable to web attacks. In this study, a CNN model with hyperparameter tuning is proposed for detecting web attacks. This paper describes the research methodology for optimizing the CNN model for web security through hyperparameter tuning to enhance its detection accuracy. It is observed that the CNN model with hyperparameter tuning for batch size 32 achieves an accuracy of 97.16%, which is 0.71% higher than the same model without hyperparameter tuning for batch size 32 and 5.67% higher compared to the CNN model trained without hyperparameter tuning for batch size 16. This means that by tuning the hyperparameters of the CNN model, the accuracy improves significantly, nearly 6.2%.

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

References

[1] Dawadi, B.R.; Adhikari, B.;Srivastava, D.K. Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. Sensors 2023.

[2] Alaoui, R.L.; Nfaoui, E.H.Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review. Future Internet 2022.

[3] Song, X.; Zhang, R.; Dong, Q.; Cui, B. Grey-Box Fuzzing Based on Reinforcement Learning for XSS Vulnerabilities. Appl. Sci. 2023.

[4] Jeklin Harefa, Gredion Prajena, Alexander, Abdillah Muhamad, Edmundus Valin Setia Dewa, Sena Yuliandry , SEA WAF: The Prevention of SQL Injection Attacks on Web Applications, Advances in Science, Technology and Engineering Systems Journal, 2021.

[5] Devendra Kumara, Dr. Anil Kumarb, Dr. Laxman Singh, Enhance Web Application Security Using Obfuscation, Turkish Journal of Computer and Mathematics Education, 2021.

[6] Mrs. Joshi Padma. N, Dr. N. Ravishankar, Dr. M. B. Raju, N.Ch. Ravi, Advanced LSTM Attention with Hybrid Framework for Detection and Prevention of SQLI and XSS Attacks, Ymerdigital, 2022.

[7] M.N.Kavitha, V. Vennila, G.Padmapriya, A. Rajiv Kannan, Prevention Of Sql Injection Attack Using Unsupervised Machine Learning Approach, International Journal of Aquatic Science, 2021.

[8] Oluwakemi Christiana Abikoye, Abdullahi Abubakar, Ahmed Haruna Dokoro, Oluwatobi Noah Akande and Aderonke Anthonia Kayode, A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm, EURASIP Journal on Information Security, 2020.

[9] Vivek Thoutam, SQL Injection Vulnerabilities Prevention through ML IPAAS Architecture, International Journal of Novel Research and Development, 2022.

[10] Robinson, Memen Akbar, Muhammad Arif Fadhly Ridha, SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application Firewall, International Journal On Informatics Visualization, 2018.

[11] Raj Agarwal, Sumedha Sirsikar, An Efficient Technique for finding SQL Injection using Reverse Proxy Server, International Research Journal of Engineering and Technology (IRJET), 2019.

[12] ANTONÍN STEINHAUSER and PETR TŮMA, Database Traffic Interception for Graybox Detection of Stored and Context-sensitive XSS, Digital Threats: Research and Practice, 2020.

[13] Ng Yi Xuan, Julia Juremi, Nurul Husna Mohd Saad, Securing e-commerce against SQL injection, cross site scripting and broken authentication, Journal of Applied Technology and Innovation, 2021.

[14] Dr.S.Saravanan, Mrs. G Keerthana, SQL Injection Prevention Using the Metasploit Framework Forweb Application, International Journal of Advances in Engineering and Management, 2021.

[15] Chengwan He and Yue He, A Reusable SQL Injection Detection Method for Java Web Applications, KSII Transactions on Internet and Information Systems, 2020.

Received: July 22, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[16] Monali Shetty, Chirantar Nalawade, Hybrid approach for Detection and Analysis of SQL and XSS vulnerabilities, International Journal of Engineering Trends and Technology (IJETT), 2018.

[17] Khalid Akram,Gulzar Banu, Mustafa Basthikodi,Ahmed Rimaz Faizabadi, Defense Mechanism Using Multilayered Approach and SQL Injection Methods for Web Based Attacks, Journal of Emerging Technologies and Innovative Research (JETIR), 2019

[18] Manuel Leithner, Bernhard Garn, Dimitris E. Simos, HYDRA: Feedback-driven black-box exploitation of injection vulnerabilities, Elsevier, 2021.

[19] Ines Jemal, Omar Cheikhrouhou, Habib Hamam, Adel Mahfoudhi, SQL Injection Attack Detection and Prevention Techniques Using Machine Learning, International Journal of Applied Engineering Research, 2020.

[20] Adamu Bin Ibrahim, and Shri Kant, Penetration Testing Using SQL Injection to Recognize the Vulnerable Point on Web Pages, International Journal of Applied Engineering Research, 2018.

[21] K. Joylin Bala, E.Babu Raj, A. M. Anusha Bamini, XSS Attack Prevention over Code Injection Vulnerabilities in Web Applications, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019.

[22] Muhammad Noman Khalid, Muhammad iqbal, Kamran Rasheed, Malik Muneeb Abid, Web Vulnerability Finder (WVF): Automated Black- Box Web Vulnerability Scanner, I.J. Information Technology and Computer Science, 2020.

[23] Khalid Akram, Gulzar Banu, Mustafa Basthikodi, Ahmed Rimaz Faizabadi, Defense Mechanism Using Multilayered Approach and SQL Injection Methods for Web Based Attacks, Journal of Emerging Technologies and Innovative Research (JETIR), 2019.

[24] Malik Qasaimeh, Ala'a Shamlawi, Tariq Khairallah, Black Box Evaluation Of Web Application Scanners: Standards Mapping Approach, Journal Of Theoretical And Applied Information Technology, 2018.

[25] Yirui Wu , Dabao Wei, Jun Feng, Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey, Security and Communication Networks Volume 2020, Article ID 8872923, 17 pages https://doi.org/10.1155/2020/8872923.

[26] Yao Pan, Fangzhou Sun, Zhongwei Teng, Jules White, Douglas C. Schmidt, Jacob Staples, Lee Krause, Detecting web attacks with end-to-end deeplearning, JournalofInternet Services andApplications https://doi.org/10.1186/s13174-019-0115-x, (2019) 10:16.

[27] Arun kumar, M. Vishwanthi, V. Shirisha, K. Priyanka, DETECTING WEB ATTACKS WITH END-TO-END DEEP LEARNING, International Journal for Advanced Research in Science and Technology,Volume 14, Issue 12, Dec 2021.

[28] Alaoui, R.L.; Nfaoui, E.H. Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review. Future Internet 2022, 14, 118. https://doi.org/10.3390/fi14040118.

[29] Noman, M.; Iqbal, M.; Manzoor, A. A Survey on Detection and Prevention of Web Vulnerabilities. Int. J. Adv. Comput. Sci. Appl. 2020, 11, 521–540.

Received: July 22, 2025