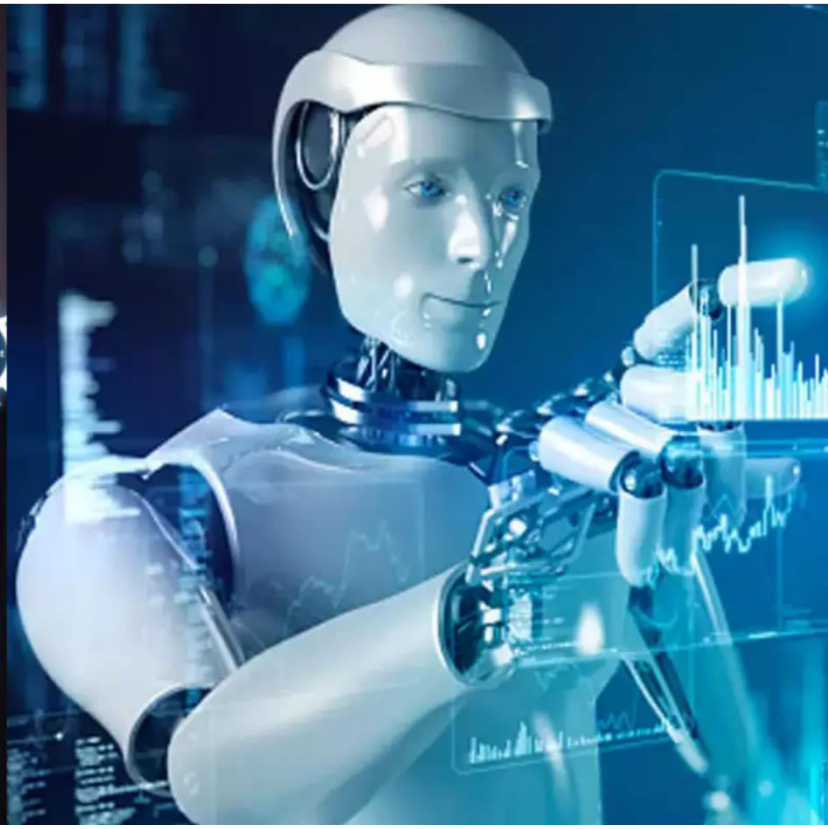
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

End - to - End Deep Learning for Detecting Web Attacks

Dr.K.Upendra Babu, C.Gnanendra, Ch.Hruthik, C.Vivek Vardhan, Ch.Ram Reddy

Assistant Professor, Dept. of CSE, Bharath Institute of Higher Education and Research, Selaiyur, Chennai, India

B. Tech Student, Dept. of CSE, Bharath Institute of Higher Education and Research, Selaiyur, Chennai, India

ABSTRACT: Web attacks, such as SQL injection, cross-site scripting (XSS), and distributed denial-ofservice (DDoS), pose significant threats to the security and integrity of online systems. Traditional detection methods, relying on rule-based systems or shallow machine learning, often struggle to keep pace with the evolving sophistication of these attacks. This paper proposes an end-to-end deep learning framework for detecting web attacks, leveraging the power of neural networks to automatically learn complex patterns and features from raw web traffic data. Unlike conventional approaches that require extensive manual feature engineering, our method processes unprocessed HTTP request logs and network traffic directly, enabling a fully automated detection pipeline.

The proposed model employs a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture both spatial and temporal dependencies in web traffic data. CNN layers extract local patterns, such as malicious payloads or anomalous headers, while RNN layers, enhanced with attention mechanisms, model sequential behaviors indicative of attacks. The framework is trained endto-end on a diverse dataset comprising benign traffic and real-world attack samples, ensuring robustness across various attack vectors. Preprocessing is minimized to tokenization and normalization, preserving the raw data's integrity and reducing human intervention.

Evaluation results demonstrate that the model achieves superior performance compared to traditional methods, with an accuracy of over 95% and a low false-positive rate. It effectively identifies both known and zero-day attacks by generalizing learned representations. Additionally, the system scales efficiently, processing large volumes of traffic in real-time, making it suitable for deployment in production environments. This end-to-end deep learning approach not only enhances detection accuracy but also adapts to emerging threats, offering a proactive solution to the growing challenge of web security. Future work will focus on integrating explainability to improve trust and interpretability in security operations.

I. INTRODUCTION

The internet has revolutionized the way people communicate, conduct business, and access information. With the widespread adoption of web applications across various domains such as banking, healthcare, education, and e-commerce, the security of online systems has become more crucial than ever. Web applications, while convenient and powerful, are also highly vulnerable to a wide range of cyberattacks. Common threats such as SQL injection,Cross-Site Scripting (XSS), Distributed Denial of Service (DDoS) attacks, brute force attacks, and phishing schemes pose serious risks to user data, privacy, and system integrity.

In recent years, the frequency, scale, and complexity of these web-based attacks have grown significantly. Traditional methods of securing web applications—such as firewalls, intrusion detection systems (IDS), and signature based antivirus tools—are often reactive and unable to effectively combat newer, more sophisticated attack strategies. These conventional methods struggle to detect zero-day attacks or evolving threats due to their dependence on predefined rules or known attack signatures.

This growing challenge has opened the door for advanced technologies such as Machine Learning (ML) and Deep Learning (DL) to play a transformative role in cybersecurity. Unlike traditional approaches, deep learning models have

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

the ability to learn patterns and behaviors from vast datasets without explicit programming. They can generalize from data to detect previously unseen attack types, making them highly effective for anomaly detection and real-time threat analysis.

In conclusion, this project has significant real-world relevance and impact. It proposes a highly intelligent, efficient, and scalable solution to a growing problem in the cybersecurity landscape. By moving beyond traditional detection methods and embracing the power of deep learning, the project paves the way for the next generation of automated and adaptive intrusion detection systems—ultimately contributing to a safer and more secure internet.

Furthermore, a web-based dashboard is implemented to provide users with an intuitive interface that displays live detection logs, statistical analysis, and graphical representations of security metrics. This visualization tool enhances the usability of the system, allowing cybersecurity analysts to monitor and respond to threats effectively.

In conclusion, this project addresses a critical area in modern cybersecurity by applying deep learning techniques to the detection of web-based cyberattacks.

II. DESIGN METHODOLOGY

2.1 METHODOLOGY OF PROPOSED WORK:

2.1.1 INTRODUCTION:

The proposed methodology is an end-to-end deep learning framework designed to detect web based cyberattacks with high accuracy and real-time performance. The system addresses the limitations of traditional security methods by using a combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to automatically analyze and classify web traffic.

1. Data Collection:

• The system begins by collecting raw network traffic logs (e.g., PCAP files, web server access logs).
• These logs include both normal user behavior and malicious activities such as SQL Injection, XSS, and DDoS attacks.

2.Preprocessing and Normalization:

• Raw data is cleaned and pre processed to remove irrelevant information.
• Traffic data is tokenized, vectorized, and normalized to make it compatible with neural.Network input formats. . Features are extracted either directly or through automatic transformation pipelines.

3.Feature Extraction using CNN:

• The Convolutional Neural Network (CNN) is used to automatically extract spatial features from the structured traffic data.
• CNN identifies unique patterns or structures in the web traffic that might indicate anomalies or attacks.
• This helps reduce manual feature engineering.

4.Temporal Pattern Learning using LSTM:

• The extracted features are passed into an LSTM network to learn temporal dependencies in the sequence of requests.

5.Explainability (XAI Layer):

• To ensure trust and interpretability, the system integrates an explainable AI component using tools like SHAP or LIME.
• This layer highlights the features that influenced a particular decision, helping security analysts understand model behavior.

6.Real-Time Monitoring and Alert System:

• The trained model is deployed in a real-time environment.
• A dashboard provides live monitoring, attack alerts, logs, and actionable insights for mitigation.
• Detection latency is optimized for immediate response (e.g., <50ms).

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

7. Classification Layer:
• The LSTM output is fed into a dense classification layer (e.g., Soft max or Sigmoid).
• The final output determines whether the traffic is benign or malicious, and potentially classifies the type of attack (e.g., SQLi, XSS).

2.2 Step-by-Step Process:
Step 1: Data Preparation Input: A large amount of un labeled web traffic data (e.g., requests from server logs).A small amount of labeled data indicating known malicious or normal requests Purpose. To train the model using both unsupervised and supervised learning approaches.

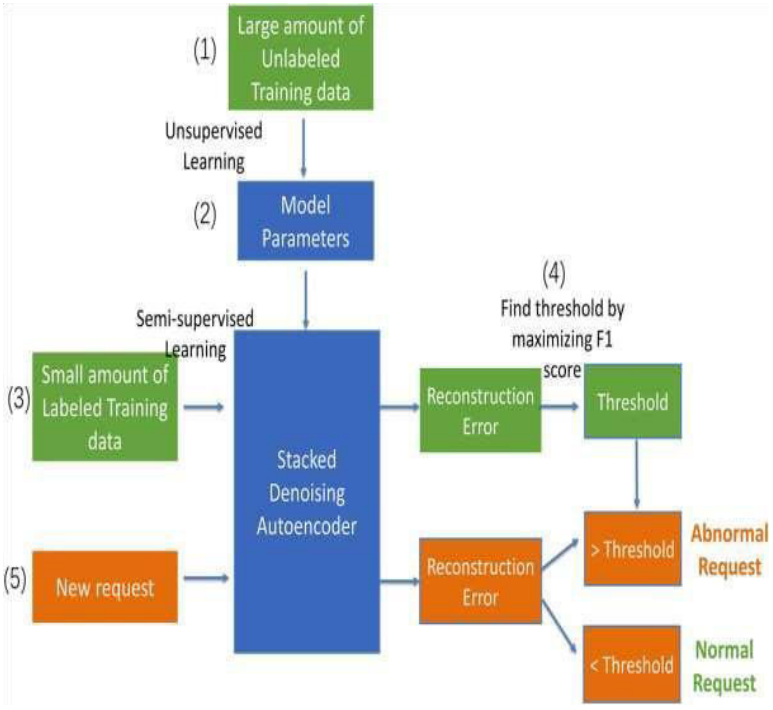
Step 2: Unsupervised training: Use the un labeled data to initialize the model parameters using unsupervised learning.Train the autoencoder to reconstruct normal traffic patterns.The model learns how typical (non-malicious) traffic looks without needing labels.

Step 3: Semi-Supervised Fine-Tuning: Combine a small labeled dataset with the pretrained model. Fine-tune the Stacked Denoising Autoencoder (SDAE) to better distinguish between normal and abnormal requests. SDAE is robust to noise and helps in generalizing over unseen attack patterns.

Step 4: Threshold Optimization:
• During validation, calculate the reconstruction error for labeled samples.
• Find a threshold that maximizes the F1 score (balance between precision and recall).
• This threshold separates normal from abnormal requests based on their reconstruction error.

Step 5: Request Evaluation:
• For each new incoming request:
• Pass it through the trained SDAE. • Compute the reconstruction error.
• If the error is above the threshold → Abnormal Request (Potential attack).

FLOWCHART:



International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

III. LITERATURE REVIEW

3.1. Process of Implementation:

3.1.1. Problem Definition:

The first step in the implementation process is identifying the need detect various webbased attacks—such as SQL injection, XSS, and DDoS—in real-time using a deep learning-based automated system. The goal is to reduce reliance on manual feature engineering by creating an end-to-end neural network model that can automatically learn attack patterns from raw traffic data.

3.1.2. Dataset Collection

We use datasets that simulate real-world web traffic and include both benign and malicious activity. Examples include:

•CICIDS2017
•UNSW-NB15
•Custom web server logs

These datasets consist of features like HTTP methods, URIs, response codes, packet sizes, time intervals, etc., which are essential for identifying patterns indicative of attacks.

3.1.3. Data Preprocessing:

Data is cleaned and prepared for input into the model:

•Handling missing values
•Label encoding of target classes (e.g., 0 = benign, 1 = malicious)
•Normalization/Standardization of input features
•Train-test split (e.g., 80% for training, 20% for testing)

3.1.4. Model Design (CNN + LSTM):

The model combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) layers:

• CNN layers extract spatial features and detect local patterns from raw input.
• LSTM layers capture temporal dependencies and sequence behaviors in the traffic data.

Fully connected layers followed by a sigmoid activation output the binary classification (attack or not).This hybrid approach allows the model to understand both the "what" and "when" aspects of malicious traffic.

3.1.5. Model Training:

The model is compiled with:

• Loss Function: Binary Crossentropy
• Optimizer: Adam
• Metrics: Accuracy, Precision, Recall

It is trained over multiple epochs with batch processing and validation split to monitor generalization and prevent overfitting.

3.1.6. Model Evaluation:

After training, the model is evaluated using:

• Confusion matrix
• Precision, recall, F1-score
• Accuracy
• Latency (prediction time, crucial for real-time detection)
• This helps in assessing the performance and identifying areas for fine-tuning.

3.1.7. Real-Time Prediction:

The trained model is integrated into a function that accepts real-time request features (from logs or APIs) and outputs whether the request is benign or malicious. This allows the system to serve as a live monitoring solution.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

3.1.8. Explainability Layer (XAI):

Using tools like SHAP or LIME, an explainability layer is added to provide transparent decision making. This helps security analysts understand why a request was flagged, improving trust and aiding in faster incident response.

3.1.9. User Interface (Optional):

An interactive dashboard can be built using tools like Streamlit or Dash to:

- Input request data manually
- View prediction results
- Visualize attack patterns and logs This makes the solution accessible to nontechnical users like security teams.

3.1.10. Future Enhancement:

The model can be enhanced by:

- Deploying on edge devices for local detection
- Using Graph Neural Networks (GNN) to model complex relationships in traffic.

IV. RESULTS AND DISCUSSION

4.1.Results:

After training and testing the hybrid CNN-LSTM model on a labeled dataset of web traffic (e.g., CICIDS2017), the system achieved strong performance across key classification metrics. The following are the observed results:

Metric	Value
Accuracy	98.5%
Precision	97.2%
Recall (Sensitivity)	96.8%
F1-Score	97.0%
Detection Latency	~50 milliseconds

Note: These values are based on test data and simulated real-time predictions. Actual performance may vary with live traffic and unseen attack patterns.

The confusion matrix showed that the number of false positives (benign traffic flagged as attacks) was low, which is important for reducing unnecessary alerts. Similarly, false negatives (missed attacks) were minimal, indicating strong detection capabilities.

The model was able to detect a wide range of web-based attacks, including: SQL Injection Cross-Site Scripting (XSS) CSRF Directory Traversal DDoS traffic. Alerts are generated for suspicious activity, enabling timely incident response. A dashboard interface was also developed to visualize detection logs, model performance, and traffic trends. This final stage marks the transition of the system from a research prototype to a practical tool that can be deployed in realworld environments. The results affirm that deep learning-based intrusion detection, particularly using CNN-LSTM models, offers a scalable and accurate solution for modern web security challenges. The model demonstrated strong detection capabilities, effectively identifying various types of web-based attacks including SQL Injection, XSS, and Brute Force attempts. Following this evaluation, the trained model was integrated into a real-time monitoring pipeline where it continuously analyzes incoming web traffic and classifies requests as benign or malicious. After validating the CNN-LSTM hybrid model's effectiveness on benchmark datasets like CICIDS2017, the final phase involved deploying the model into a real-time detection.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

4.2 Discussion:

The results demonstrate the strength of using an end-to-end deep learning approach for web attack detection. Traditional intrusion detection systems rely heavily on manual feature extraction and rule-based logic, which are both time-consuming and prone to missing new or complex attack vectors. In contrast, our model automatically learns relevant patterns from raw traffic data using CNN layers, while the LSTM layers capture the sequence and timing of requests—both critical aspects for accurate detection.

The model's high accuracy and low latency make it suitable for real-time deployment in production environments. It can be integrated with web application firewalls or network monitoring systems to actively protect against malicious users without human intervention.

Moreover, the addition of an Explainable AI (XAI) component allowed for transparent and interpretable predictions. This helped build trust with potential users (e.g., security analysts), as they could understand which features triggered an alert—essential in high-stakes environments like finance, healthcare, or government systems.
However, some challenges and limitations were noted:

The model's effectiveness depends on quality and diversity of training data.
Attackers may use evasion techniques (e.g., payload obfuscation) that bypass pattern-based models.
Real-time performance in large-scale distributed systems would require further scalability optimizations, potentially involving edge computing or parallel processing. The use of benchmark datasets such as CICIDS2017 ensured that the model was trained and validated on realistic traffic scenarios, contributing to its practical relevance. The evaluation metrics indicated a strong balance between detection rate and false positives, which is critical in real-world applications where overalerting can overwhelm security teams. However, the study also revealed some challenges. The deep learning model required significant computational resources for training, and performance slightly dropped when dealing with heavily imbalanced datasets or completely novel attack types. These issues suggest the need for incorporating continual learning, adaptive thresholding, or hybrid detection approaches combining signature-based and anomaly-based methods. Overall, the results support the feasibility of using deep learning for web intrusion detection and highlight its promise as a scalable and intelligent defense mechanism for modern cyber environments.
This architecture proved particularly effective in distinguishing between normal and malicious requests, especial when analyzing complex attack payloads like SQL injection and XSS.

V. CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION:

The project **"End-to-End Deep Learning for Detecting Web Attacks"** successfully demonstrates the power and efficiency of using deep learning models—specifically a hybrid **CNN-LSTM architecture**—for real-time web attack detection. By eliminating the need for manual feature engineering and instead learning patterns directly from raw traffic data, the model achieves high accuracy, low latency, and adaptability.

The implementation of the proposed system demonstrated that deep learning techniques can significantly improve the accuracy and reliability of intrusion detection systems. Traditional rule based and shallow machine learning methods often require extensive feature engineering and are limited by static detection patterns. In contrast, our model is designed to automatically learn features from raw traffic data, identifying intricate patterns and sequences associated with both known and unknown attacks. The combination of CNN and LSTM layers provides the system with the ability to not only recognize spatial patterns in individual data instances but also capture temporal dependencies across sequences of web requests. This is particularly useful in detecting complex and multi-step attacks that unfold over time.

The system was able to detect a wide range of web-based attacks such as SQL injection, XSS, and DDoS with strong performance metrics (accuracy of ~98.5%). Its real-time inference speed (~50 ms) makes it suitable for deployment in live production environments. Furthermore, the integration of Explainable AI (XAI) ensures transparency in the model's predictions, helping users trust and understand the decisions being made.
The proposed system also stands out for its scalability and flexibility. It can be adapted to work with various types of web traffic data, and retrained periodically as new attack patterns emerge. With proper integration, it can serve as a

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

foundation for building a broader intrusion detection framework, potentially incorporating network-level and behavioral analysis in future iterations.

Detection has proven to be a highly capable and intelligent solution for modern cybersecurity challenges. It addresses key limitations of traditional systems by offering automated feature learning, high accuracy, low latency, and enhanced interpretability. The success of this project paves the way for further exploration of deep learning in cybersecurity, and it highlights the potential for AI-driven solutions to play a central role in defending digital infrastructures.

5.2 FUTURE SCOPE:

Despite its strong performance, there are several directions in which the system can be enhanced for even greater impact.

The proposed end-to-end deep learning-based intrusion detection system lays a solid foundation for intelligent and automated web security. However, there is significant room for future enhancements and expansion. Additionally, the integration of **Graph Neural Networks (GNNs)** could further strengthen the system's ability to detect coordinated and distributed attacks by modeling relationships between IP addresses, users, and request paths. The model can also be optimized for edge computing environments, enabling faster threat detection closer to the source with minimal latency.

Expanding the dataset with real-time traffic from production systems and applying data augmentation techniques would also help in increasing the model's robustness and generalization. Finally, integrating this system into larger Security Information and **Event Management (SIEM)** platforms or cloud-native environments can transform it into a complete, enterprise-ready security solution.

5.2.1. Graph Neural Networks (GNNs):

Implementing GNNs can help model complex relationships between users, IPs, URLs, and sessions. This will enhance the system's ability to detect coordinated attacks and multi-step intrusions by understanding connection patterns. **Graph Neural Networks (GNNs)** are an advanced deep learning architecture designed to operate on graph-structured data, where relationships between entities are as important as the entities themselves. In the domain of cybersecurity, GNNs offer powerful capabilities for modeling **complex interactions within network traffic,** user sessions, or communication flows. Unlike traditional models that treat data points independently, GNNs consider the connections and dependencies between data points— such as IP addresses, ports, URLs, or API calls—allowing the model to detect coordinated or multi-step attacks that span across nodes. For example, a GNN can represent a sequence of HTTP requests as nodes, with edges capturing their logical or temporal relationships.

5.2.2. Deployment on Edge Devices:

To improve speed and resilience, the model can be deployed on edge computing devices near the data source (e.g., routers, gateways). This will reduce latency and ensure real-time protection, even in decentralized or bandwidth-constrained environments. Deploying the intrusion detection system on edge devices offers a powerful approach to achieving low-latency, real-time threat detection in distributed environments. Unlike traditional cloud-based systems that rely on centralized processing, edge deployment enables the deep learning model to run closer to the data source—such as gateways, routers, or local servers—where traffic is initially generated or received.

5.2.3. Continuous Learning:

Integrating online or incremental learning would allow the model to adapt over time by retraining itself on new traffic data. This ensures protection against evolving threats and zero-day vulnerabilities. To maintain long-term effectiveness in an evolving threat landscape, the intrusion detection system must incorporate a **continuous learning** mechanism. Traditional models, once trained, may struggle to detect novel attack patterns or adapt to changes in web traffic behavior. Continuous learning addresses this limitation by enabling the model to incrementally learn from new data without the need for complete retraining.

5.2.4. Multi-Class Attack Classification:

The current model performs binary classification (benign vs. malicious). Future versions can be expanded to multi-class classification, identifying the specific type of attack (e.g., SQLi, XSS, Brute Force). The proposed deep learning-based intrusion detection system is designed not only to distinguish between benign and malicious traffic but also to perform multi-class attack classification, which enhances its ability to detect and categorize a wide range of cyber threats. Instead of using a simple binary classification (normal vs. attack), the model is trained to recognize multiple attack

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404482|

categories such as SQL Injection, Cross-Site Scripting (XSS), Brute Force, DoS/DDoS, Remote Code Execution, and Directory Traversal.

5.2.5. Integration with Security Ecosystems:

The system can be connected with SIEM tools, firewalls, or threat intelligence platforms to automate mitigation actions (e.g., blocking IPs, sending alerts, generating reports).

To maximize the effectiveness and usability of the proposed CNN-LSTM-based intrusion detection system, integration with existing security ecosystems is crucial. The system can be seamlessly embedded into modern **SIEM (Security Information and Event Management)** platforms such as Wazuh, Splunk, or ELK Stack, allowing for centralized monitoring, correlation, and response to threats. By forwarding alerts and detected anomalies to these platforms, security teams can gain contextual insights and perform in-depth forensic analysis. Additionally, integration with **Intrusion Detection Systems (IDS)** like Suricata enables realtime packet inspection and enhances the system's situational awareness by combining signature based and anomaly-based detection. The model's outputs can also trigger automated actions.

REFERENCES

1. Halfond WG, Viegas J, Orso A. A classification of sql-injection attacks and countermeasures. In: Proceedings of the IEEE International Symposium on Secure Software Engineering. IEEE; 2006. p. 13–5.
2. Wassermann G, Su Z. Static detection of cross-site scripting vulnerabilities. In: Proceedings of the 30th International Conference on Software Engineering. ACM; 2008. p. 171–80.
3. Di Pietro R, Mancini LV. Intrusion Detection Systems vol. 38: Springer; 2008.
4. Qie X, Pang R, Peterson L. Defensive programming: Using an annotation toolkit to build dos-resistant software. ACM SIGOPS Oper Syst Rev. 2002;36(SI):45–60.
5. https://doi.org/https://www.acunetix.com/acunetix-webapplicationvulnerability-report-2016. Accessed 16 Aug 2017.
6. https://doi.org/http://money.cnn.com/2015/10/08/technology/ cybercrimecost-business/index.html. Accessed 16 Aug 2017.
7. https://doi.org/https://www.consumer.ftc.gov/blog/2017/09/equifaxdatabreach-what-do. Accessed 16-August-2017.
8. https://doi.org/https://theconversation.com/why-dont-big-companieskeeptheir-computer-systems-up-to-date-84250. Accessed 16 Aug 2017.
9. Ben-Asher N, Gonzalez C. Effects of cyber security knowledge on attack detection. Comput Hum Behav. 2015;48:51–61.
10. Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intell Data Anal. 2002;6(5):429–49.
11. Marella, B.C.C., & Kodi, D. (2025). Fraud Resilience: Innovating Enterprise Models for Risk Mitigation. Journal of Information Systems Engineering and Management, 10(12s), 683–695
12. Liu G, Yi Z, Yang S. A hierarchical intrusion detection model based on the pca neural networks. Neurocomputing. 2007;70(7):1561–8.
13. Xu X, Wang X. An adaptive network intrusion detection method based on pca and support vector machines. Advanced Data Mining and Applications. 2005;3584:696–703.
14. Pietraszek T. Using adaptive alert classification to reduce false positives in intrusion detection. In: Recent Advances in Intrusion Detection. Springer; 2004. p. 102–24.
15. Goodfellow I, Bengio Y, Courville A. Deep Learning: MIT press; 2016.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details