International Journal of Research in Science and Technology

Volume 13 Issue 01 January-March 2026

ISSN 2394 - 9554

An Attention-Enhanced CNN–BiLSTM Framework for Zero-Day Exploit Detection in Network Traffic

Vigneswari V[1], Velsabarigiri K[2] and R Manimegalai[3]

Department of Computer Science and Business Systems

PSG Institute of Technology and Applied Research, Coimbatore, India

*E-mail: drrm@psgitech.ac.in

Abstract

Zero-day exploits have a very significant threat to modern computing systems as they are focused on exploiting the unknown vulnerabilities of the computing system. Conventional computing systems are unable to detect such attacks as the patterns of the attacks keep changing, and the aforementioned techniques of detection are rule-based or signature-based. This paper presents a deep learning-based approach that can detect the existence of zero-day exploits on network traffic using a hybrid Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) network that is enhanced by the attention mechanism. The proposed approach emphasizes the key features that are relevant to the abnormal behavior and effectively models the spatial as well as the temporal patterns. Techniques such as threshold optimization, class-weighted focal loss, injection of gaussian noise, and dimensionality reduction are utilized to handle the issue of imbalance in the classes. Accuracy, recall, and a low value of the positive predictive rate are demonstrated in the experimental results on the publicly available network traffic dataset of the logistics network, and the proposed approach is thus demonstrated to be fitting and effective for proactive detection of zero-day exploits.

Keywords: *Zero-day Exploit Detection,, CNN-BiLSTM, Attention Mechanism,, Anomaly Detection, Logistics Network Traffic*

1. INTRODUCTION

In recent years, cyber threats have increased significantly because of the rapid growth of digital technologies, computer networks, and cloud-based services. Organizations often depend on their hardware and software infrastructures for storing sensitive data and performing day-to-day operations. Since everything depends on hardware and software infrastructures, attackers look for weaknesses in these systems. Zero-day exploit is an attack which poses a serious threat to the computing environment in which the attackers exploit the vulnerabilities in the systems that the developers and security team are not aware of at the time of attack[2]. Traditional security mechanisms such as firewalls, antivirus software, and signature-based intrusion detection systems primarily depend on known attack patterns. Thus, all these mechanisms are effective only after an attack has been discovered and documented. However, standard systems become useless in zero-day attacks as they occur before any patch or signature is prepared[9]. Due to this limitation, zero-day attacks often cause significant damage and can remain unnoticed for long periods of time. The impact of zero day exploits are demonstrated by a number of occurrences in real world scenarios. Case studies on Log4j vulnerability and HAFNIUM assaults on Microsoft Exchange servers depict the exploitation of un-discovered vulnerabilities by attackers to install malware, steal data, and spread ransomware across thousands of companies worldwide. These examples

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

quite clearly bring out the point that modern zero-day threats cannot be handled by reactive security measures alone[3].A lot of research has been carried out on machine learning-based intrusion detection systems, to get over the deficiencies of signature-based approaches. For the identification of unknown assaults in network flows, a hybrid learning framework combining both the supervised and unsupervised techniques was proposed and the results reflected low false detection rate along with high accuracy of detection, which established the efficacy of these methods for the detection of zero-day attacks[6]. Other than the learning-based approaches, honeypots, honey tokens, and camouflaging are some of the deception-based security approaches that are being suggested in the literature for managing zero-day attacks[8]. These approaches attempt to learn more about the knowledge of how attacks are carried out by misleading attackers away from actual systems. As proposed by the researchers, the integration of deception approaches with machine learning models will help improve security effectiveness and early detection[4]. In another direction, advanced deep learning architectures including auto encoders, attention mechanisms, and hybrid ensembles have also been presented in the recent works to improve the performance of zero-day exploit detection. These approaches focus on reducing false positives along with improving adaptiveness and, in turn, detecting unidentified attack patterns much more effectively than previously used approaches [5]. The motivation for an effective and adaptive zero-day detection framework that is accurate and is capable of identifying unknown attacks in real time leads to the design of the proposed study, and it aims at improving the performance of zero-day attack detection by designing an intelligent method for detection.

2. Literature Survey

Zero-day attacks have been a serious concern in cybersecurity because they exploit unknown vulnerabilities that have no existing patches or signatures. Early research work included explaining the concept of zero-day threats in its relative simplicity. Alongside discussing essential security practices such as Data Execution Protection (DEP) and Address Space Layout Randomization (ASLR), the topic of zero-day attacks also covered how these attacks are performed by exploiting malware and worms to target software deficiencies [1]. However, most research work does not use an automated solution that needs to be more applicable in real-time scenarios and remains mostly theoretical.

Various studies focused on zero-day attacks with case study-based methodologies to understand the detection and prevention strategies. These works underlined early-bound detection, awareness of assault, and preventive security measures considering different real-world assault scenarios [2]. Even though these analyses provide useful insights regarding the security procedures and attack behavior, they do not suggest certain implementation frameworks or learning-based detection models that might be used for automated intrusion detection.

Other research focused on real-life cases involving zero-day occurrences such as the Log4j vulnerability and the attack by HAFNIUM. The cycle of zero-day attacks, attack architectures, as well as common defense systems such as behavioral analytics, intrusion detection systems, and intrusion prevention systems was discussed in the research [3]. The proposed solutions in these researches can be mainly reaction-based and lack predictive features, even though they provide a good comprehension of the distribution process involving zero-day attacks.

There has been an indication of the use of machine learning algorithms to provide an appropriate solution for the disadvantages associated with traditional strategies for intrusion detection. Using network flow data to detect anomalies, a combination of supervised and unsupervised algorithms has been shown to improve sensitivity to intrusions and reduce the rate of false detections [4]. Though very effective, these

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

models are difficult to use in real-time because of the associated complexities and various steps involved in their processes.

Recent propositions have advanced deep learning models for zero-day exploit detection, featuring autoencoders, attention mechanisms, and hybrid ensemble architectures. These models could achieve very high accuracy with low false positive rates, especially by further improving the feature extraction process and the capability to learn complex attack patterns[5]. However, these models are very complex and resource-consuming, hence challenging to understand; this, in turn, severely limits their scalability and practical applicability in real-world systems[7].

For the improvement of zero-day attacks detection, specifically for smart environments, certain works explored the focus on interoperable AI, as well as explainable AI. This focus was on building trust for AI-driven decisions, collaboration for security systems, as well as transparency [6]. The approach aims to increase explanations for AI systems, as well as integrations at the system level, with reduced focus on real-time detection and feature-level learning.

A methodology for the identification of privilege escalation attacks and their prevention through the use of machine learning was proposed by Santhi and Sundari in their research paper [6]. This paper highlights the need to make use of automated prediction tools, which is quite essential to reduce the risk of illegal access. This paper supports the idea that machine learning can be used to improve security in cloud environments. However, the methodology also has the limitation of relying on the availability of annotated log data. Privilege escalation attacks are quite rare compared to normal log data, which could lead to an imbalance. The paper also shows that there is a need for more advancements to ensure that the system remains reliable over time.

It becomes clear from the available literature that many approaches are either too complex to be implemented in real-life scenarios, lack advanced automation, and totally depend on reactive analysis. The proposed research work aims to present an optimized deep learning approach that aims to remove false positives, detect attacks in real-time with feasible computing requirements, and successfully detect unknown threats in order to overcome these limitations.

2.1 System Description

Zero-day exploits are cyber attacks that take advantage of unknown software vulnerabilities. There are no patches or signatures for which these vulnerabilities can be identified, as the developers are not yet aware of the existence of these vulnerabilities. Such attacks have proven ineffective for traditional or conventional methods for protecting networks, firewalls, or signature-based intrusion detectors. The threats posed to network or system security, mainly the stealing of sensitive information, system compromise, as well as service disruption, have prompted the need for a more intelligent system capable of pinpointing the zero-day exploit based upon the analysis of unusual network activity.

The aim of this research is to apply machine learning and deep learning concepts to build and implement an intelligent system for zero-day exploit detection. As an approach to identify unusual and unknown patterns of attacks, this system focuses on network traffic data. This work does not consider host-level and hardware-related security mechanisms and is limited to zero-day network-based detection. The objective of this model is to apply real-time detection capability in real-life scenarios for cybersecurity. It is evaluated on a publicly available dataset.

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

The main objectives of this work are:

To explore the impact of Zero-Day attacks on cybersecurity systems.

To detect attacks through the preprocessing of network traffic data.

To build a smart model which is able to detect unidentified zero-day exploits.

To reduce false positives and improve the accuracy of detection.

To test the system by common evaluation criteria like accuracy, precision, recall, and F1 score.

This study uses a dataset obtained from Kaggle, containing network traffic records classified as malicious or normal. The dataset includes several network flow features at the connection and packet levels. It contains both attack and regular traffic that helps test and train a detection model. It is also preprocessed with data cleaning, normalization, feature selection, and managing class imbalance before model training to improve model performance.

The problem of detecting zero-day exploits proves to be challenging in several respects. To begin with, since no signatures are known in the case of zero-day attacks, traditional approaches will not be very helpful. Moreover, the data obtained from the networks turns out to be highly imbalanced, with very less instances pertaining to the attack instead of the usual ones. In addition, the purpose of the attackers turns out to be simulating normal activities so that they will not be detected, thus causing more instances of false positives.

2.2 Proposed Methodology of the Detection System

The process for implementing the proposed zero-day exploit detection system will be elaborated in this section. Preprocessing, feature engineering, combining the model, model training, and model assessment are all included in the overall procedure. Care is taken in every step to increase detection accuracy while dealing with the challenges brought about by zero-day attacks.

Network traffic flow data forms the dataset on which the research has been carried out. This data was collected from Kaggle. A connection in the network is denoted by each row in the data, and there are some numeric features in this data related to packets and the time of the connection and protocol details. This learning model can differentiate between attack and normal data as there are both malicious and normal connections in the data.To identify the number of samples, types of features, and class distribution, the data set is first considered. The data set is very imbalanced, as the number of malicious instances is less than normal instances due to the rarity of zero-day attacks. Then, the data set is split into the training set and test set to perform an unbiased evaluation on the proposed model.

To ensure that the data is of good quality and to enhance efficiency in learning, data preprocessing occurs. Missing information that can lead to inappropriate learning is removed. Each attribute is then converted to a numerical format that can be employed to train the model.Scaling techniques are utilized in order to facilitate normalization, as the features of the network traffic may have varying value ranges. This made sure that the features hold equal weight and those with higher magnitude do not dominate others. In addition to that,the preprocessing step helped to increase the convergence rate for models.

Feature engineering plays an important role in detecting zero-day exploits. In our research, feature selection helps alleviate the problem of unnecessary and redundant feature sets being analyzed.

Correlation Filtering: Correlated, as well as unnecessary, features were eliminated in order to avoid repetition using a correlation matrix.

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

Final Feature Set: The final feature set that will be considered comprises the following features: BTC, USD, Netflow_Bytes, Port, Payload Size, Clusters, Error Code, Anomaly Score.

Practical Relevance: These characteristics can be used to distinguish benign and exploit traffic based on traffic volume, packet patterns, protocol errors, clustering characteristics, and anomaly scores.

Benefits: There is less noise, ease of model interpretation, and convergence speed because training focuses on high-impact features.

An attention technique is employed to enhance the CNN-BiLSTM model at the system's core. The extraction of spatial features, learning sequential patterns, and temporal feature weighting were supposed to be combined in one process by the model.

Input Layer and Gaussian Noise Layer: This layer enhances the robustness of the network against intermittent fluctuations in the actual network values with the help of Gaussian noise.

1D Convolution Layer: This layer emphasizes significant connections in the input sequence by extracting local feature patterns and performing the roles of a filter.

Bidirectional LSTM Layer: Helps the model to understand the context by identifying the relationships in both forward and backward sequences.

Attention Layer: This layer is very important as it uses attention to highlight the important time steps and features in the input sequence. As a result, the layer can

Dropout and Batch Normalization: Through disrupting the co-adaptability between neurons and normalized activations, both Dropout and Batch Normalization reduce overfitting.

The entire workflow of the research work is explained in Fig.1.

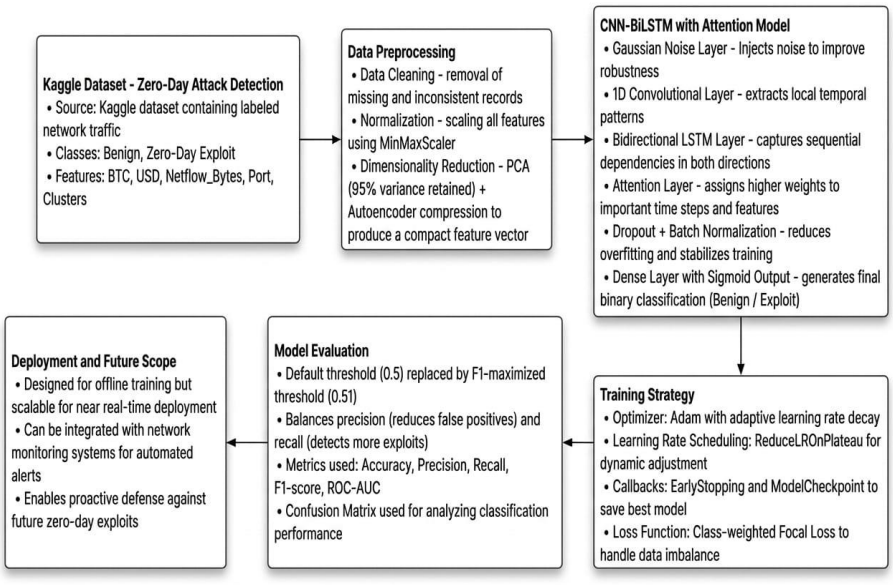

Fig.1 - Workflow Diagram

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

The Adam optimizer was used with adaptive learning rate decay. When the loss stopped improving in the validation dataset, a learning rate scheduler came into play, automatically reducing it by a factor to ensure efficient convergence.

In order to prevent overfitting, the training process was carried out with:

EarlyStopping: Stopped training when validation loss stopped improving.

ModelCheckpoint: The best model according to validation criteria was saved.

Class-weighted Focal Loss: Made the minority class-exploit samples-get greater weights in order to deal with an excessive class imbalance.

As the default threshold of 0.5 is used, a slight tendency towards the benign class is favored. Then, the optimization of the value of the threshold to maximize the F1-score is done, providing a value of 0.51 for the threshold, which tries to strike a balance between the probability of identifying more exploits and the possibility of revealing them.

The results on the test set were evaluated on the following criteria:

Accuracy, Precision (regularity of positive predictions), Recall, F1 - score (Harmonic Mean of Recall & Precision), ROC-AUC - Overall Ability to Distinguish Between Classes, Confusion Matrix (Distribution of TP, TN, FP, and FN for insight).

The test determined that the model was near perfection with regard to the ROC-AUC and Accuracy metrics.

3. Experimental Results And Analysis

The Kaggle dataset for zero-day exploit was employed to train the CNN-BiLSTM model with attention for 40 epochs with an optimal size for the batch size. To avoid the model getting stuck, the ReduceLROnPlateau scheduler along with the Adam optimizer that has the ability to adapt the learning rate was employed for the model's training process. To ensure that the model does not suffer from the problem of overfitting, along with the use of early stopping, the ModelCheckpoint callback function was successfully utilized.

The test set held back was employed for evaluating the feasibility of the proposed model. The performance metrics included precision, recall, F1 score, ROC-AUC score, and confusion matrix. The capability of effective exploit detection with a good tradeoff between precision and recall has also been ascertained.The experimental results of the work are shown in Fig.2.

	Metric	Value
0	Accuracy	95.600000
1	ROC–AUC	0.992400
2	F1-Score	0.942300
3	Precision	0.943100
4	Recall	0.941400

Fig.2 - Model Evaluation Metrics

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

Just a tiny fraction of well over 19,000 test samples was mistakenly labeled (314 were false positives, and 324 were false negatives), as indicated in the confusion matrix, which showed correct predictions on 8707 benign and 5205 exploitation samples. Such a system can obviously differentiate well between beneficial and harmful communication.

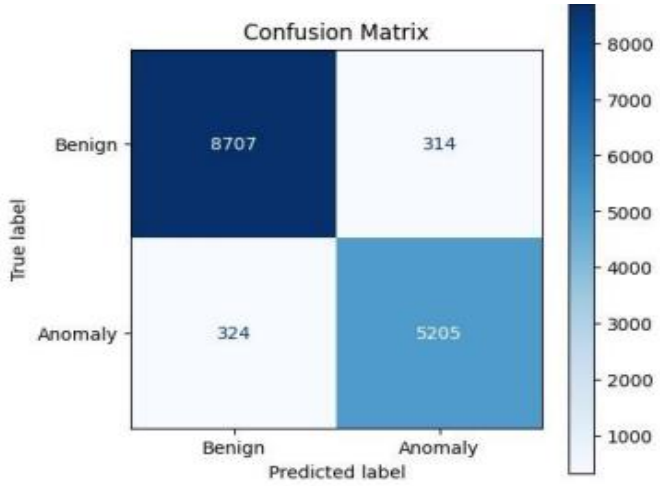

Fig.3.1 - Confusion Matrix

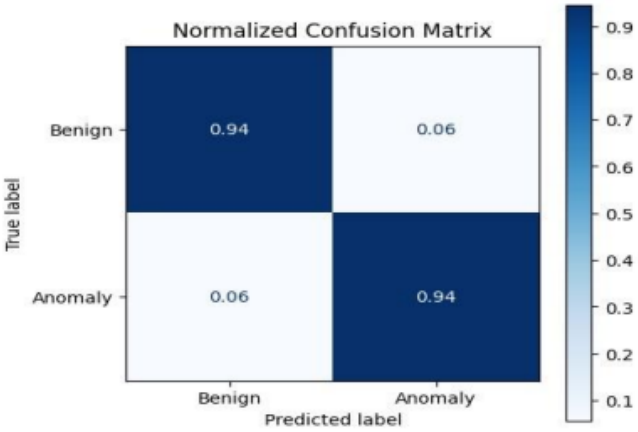

Fig.3.2 - Normalized Confusion Matrix

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

The model's confusion matrix and normalised matrix are shown in Fig.3.1 and Fig.3.2. As shown in the Fig.3.1, the model features very low false positives (3%) and has excellent detection capability: it correctly classifies 94% of abnormalities and 97% of benign traffic. Strong recall and good generalization to unseen exploit patterns are supported by the fact that only 6% of anomalies are missed. This performance showcases how selected architecture and threshold tuning successfully balance the decrease of false alarms with the increase of covered detection.However, since there is a problem of class imbalance, there is a slight bias introduced in the typical sigmoid threshold value of 0.5. The issue was resolved by incorporating an F1 maximization technique to obtain the optimal threshold. To optimize the performance, the optimal threshold for the sigmoid function was adjusted to 0.510 to maximize the performance of recall. Finally, to avoid overloading the emergency team in the deployment environment, the solution was made accurate and practically possible

In spite of criticisms that deep-learning models are opaque, making them difficult to interpret, having an attention component helped. It was possible to emphasize key attributes or steps in time that would help to draw attention to traffic patterns that were dominant in exploitive behavior. It was therefore possible to interpret what was being used to make decisions within traffic flow, which is necessary if security operations are to gain acceptance.

4. Observations and Insights

The CNN-BiLSTM-Attention model itself has been proven to be very efficient compared to the conventional method through the utilization of individual models, and also very accurate and reliable. Regularization has been an important part of ensuring there is no avoidance of potential events of overfitting, which are most likely to take place due to the very few examples utilized. Optimization of thresholds has also been very important in making sure there is a certain degree of balance in relation to the ability to detect, in contrast to potential events of the false positive, which itself demonstrates potential in being able to prove the statement of legitimacy that the claim, in relation to the concerned area of cybersecurity, aims at attaining beyond just accuracy. There is room for maneuver in relation to the potential ability to accomplish the objective of real-time zero-day vulnerability, in relation to the concerned logistics network.

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

5. Conclusions

The CNN-BiLSTM attention model is efficient enough to check whether or not the method used in the classification process of the zero day attacks on the logistics network communication traffic is efficient. The CNN-BiLSTM model is particularly treated before beginning the training process to resolve the problem of class imbalance by using the concept of class weights and a focal loss function. There is a thresholding technique used in the CNN-BiLSTM attention model to obtain a fair combination of the values of accuracy and precision metrics with a ROC-AUC value of 0.9924 and an accuracy measure of 96% against the statistics of the experiment result.

These are the benefits reaped by the pipeline.

High Accuracy Rates in Detection: For ensuring the exploit detection process is successful, accuracy, recall, and F1 value were employed.

Balanced Performance: In order to ensure there was high recall, which was very important.

Interpretability: the ability to comprehend the characteristics involved in the detection, increased the reliability of the approach as the attention part of the work was taken into account.

Although it has produced promising results, it has a whole lot of flaws in it. The question of generalizability might be an issue because this technique has been developed and validated in one specific logistics network environment. The training of this deep hybrid model is highly resource-intensive and also a heavy computation task requiring GPUs. This model has not been used in any real-time system so far. Currently, this model has been employed exclusively for offline detection purposes.

There are a few extra aspects to this research remaining.

Real-Time Integration: The approach also holds potential for utilization in real-time system security in real-time system surveillance.

Adaptive Learning: The reason for this approach being called adaptive learning is its keen emphasis on incremental learning.

Big Dataset Testing: This would involve testing it on a variety of networks and data so that this technique is able to attain scalability.

Explainability AI: This would incorporate the utilization of varied approaches to enable an overall greater level of specificity in security experts about various model detection decisions.

6. References

1.Mohammad Sayduzzaman, Anichur Rahman, Jarin Tasnim Tamanna, Dipanjali Kundu, and Tawhidur Rahman, “Interoperability and Explicable Artificial Intelligence-Based Zero-Day Attacks Detection Process in Smart Community,” Preprint Article, 2024

2.Kunwar Vaisla, “Analyzing of Zero Day Attack and Its Identification Techniques,” In Proceedings of First International Conference on Advances in Computing and Communication Engineering, 2014, pp. 11–13

3.Azheen Waheed, Bhavish Seegolam, Mohammad Faizaan Jowaheer, Chloe Lai Xin Sze, Ethan Teo Feng Hua, and Siva Raja Sindiramutty, “Zero-Day Exploits in Cybersecurity: Case Studies and Countermeasure,” Preprint Research Article, 2024, pp. 1–18

https://doi.org/10.64151/PSGCARE-19

ISSN 2394 - 9554

4.Nyia Okechukwu Cosmos and Gilbert Aimufua, “Application of Diverse Techniques for Zero-Day Management: A Review Approach,” International Journal of Research and Innovation in Applied Science, 2025, Volume X, pp. 1436–1452.

5.Ahmed A. Mohamed, Abdullah Al-Saleh, Sunil Kumar Sharma, and others, “Zero-Day Exploits Detection with Adaptive WavePCA-Autoencoder Adaptive Hybrid Exploit Detection Network,” Scientific Reports, 2025, Volume 15, Article No. 4036.

6.Almamy Touré, Youcef Imine, Alexis Semnont, Thierry Delot, and Antoine Gallais, “A Framework for Detecting Zero-Day Exploits in Network Flows,” Computer Networks, 2024, Volume 248, Article No. 110476.

7.Y. Guo, “A Survey of Machine Learning-Based Zero-Day Attack Detection: Challenges and Future Directions,” Computer Communications, 2023, Volume 198, pp. 1–10.

8.S. A. Alansary, S. M. Ayyad, F. M. Talaat, and others, “Emerging Artificial Intelligence Threats in Cybercrime: A Review of Zero-Day Attacks via Machine, Deep, and Federated Learning,” Knowledge and Information Systems, 2025, Volume 67, pp. 10951–10987.

9.N. Mearaj and M. A. Wani, “Zero-Day Attack Detection with Machine Learning and Deep Learning,” In Proceedings of Tenth International Conference on Computing for Sustainable Global Development, 2023, pp. 719–725.

https://doi.org/10.64151/PSGCARE-19