*International Research Journal of Innovations in Engineering and Technology (IRJIET)*

**ISSN (online): 2581-3048**

**Volume 8, Issue 5, pp 232-237, May-2024**

https://doi.org/10.47001/IRJIET/2024.805034

# Artificial Intelligence-Driven Penetration Testing for Wireless Networks: Enhancing Security Vulnerability Detection Using CNN Models

**[1]Mustafa Salim Mohammed Al-Saudi, [2]Kassem Hamze**

[1,2]Department of Computer and Communications, Faculty of Engineering, Islamic University of Lebanon, Wardanieh, Lebanon

***Abstract* - Adware, spyware, viruses, and other forms of malware are serious risks to people, companies, governments, and military activities. Advanced methods for vulnerability detection are required since traditional security measures frequently fall short in the face of complex and dynamic cyberthreats. In order to increase accuracy, adaptability, and scalability in discovering security vulnerabilities, this study investigates the incorporation of artificial intelligence (AI) in the design of a wireless network penetration testing system, utilizing machine learning. The dataset used was BoTNeTIoT-L01, which has over 7 million records of IoT botnet attacks. Using the Keras library, a convolutional neural network (CNN) model with layers for convolution, max pooling, and dense was created. The Adam algorithm was then used to optimize the CNN model's training process. The model's remarkable 99.46% accuracy rate in categorizing assaults indicates how well it can detect security holes and adjusts to emerging threats. The results also confirm the capabilities of artificial intelligence in enhancing cybersecurity measures and ensuring strong protection in increasingly complex wireless network environments.**

***Keywords:*** Penetration Testing, Artificial Intelligence, Vulnerability Detection, Convolutional Neural Networks, BoTNeTIoT-L01.

## I. INTRODUCTION

The spread of malware in the current digital era, including viruses, adware, and spyware, poses serious risks to people's safety as well as those of companies, governments, and military forces. According to the McAfee Labs report, the number of malware attacks is rising again [1]. In 2020, the number of new malware attacks declined for the first time since 2015. However, according to Sonic Wall's 2022 Cyber Threat Report [2], this was just a temporary dip, with malware attacks now sitting at 10.4 million per year, roughly where they were back in 2018. This led to significant financial losses totaling hundreds of billions of dollars. This concerning trend emphasizes how urgently sophisticated security measures are needed to safeguard sensitive data and important systems. Sophisticated malware is difficult to detect and mitigate using traditional security procedures, which are mostly dependent on signature and authentication mechanisms [3]. These traditional techniques frequently have trouble finding new vulnerabilities, especially in wireless networks where data flow is dynamic and adds to the complexity. Furthermore, maintaining strong security standards becomes more challenging as networks and connected devices grow faster and more numerous.

The emergence of artificial intelligence (AI), particularly in the areas of machine learning and deep learning, presents encouraging answers to these problems [4]. AI technologies have shown impressive results across a range of fields, enabling robots to carry out tasks that conventionally need human intelligence [5]. When trained on a sufficient and representative dataset, machine learning models can effectively generalize to identify new attack forms and dangers in the context of cybersecurity. The goal of this study is to create a sophisticated wireless network penetration testing system by utilizing artificial intelligence. The purpose of penetration testing is to improve the scalability, accuracy, and adaptability of vulnerability detection by looking for security holes in a system that an attacker could exploit.

## II. RELATED WORK

Numerous scholars have investigated and studied the field of intrusion detection system (IDS) in great detail. Using machine learning approaches in conjunction with IDS has become more popular in recent years as a way to improve the system's ability to thwart intrusions. In [6] the sparse autoencoder, an excellent learning algorithm to reconstruct a new feature representation in an unsupervised manner, was used to develop an intrusion detection model. After completing the pre-training phase, the detected features were fed into the Support Vector Machine (SVM) algorithm in order to enhance its ability to Intrusion detection. As a means to improve intrusion detection, a new feature selection approach was presented in [7] using the Naive Bayes (NB) algorithm with ten benchmark datasets obtained from the UCI library. The results show that the classifiers can achieve comparable classification performance by using only eight

 

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 8, Issue 5, pp 232-237, May-2024
https://doi.org/10.47001/IRJIET/2024.805034

features instead of requiring all 41 features in the dataset. In [8] the accuracy and misclassification rate of two frequently used classification methods: SVM and NB were studied. The purpose was to compare and contrast the two approaches. The NSL-KDD dataset was used and in order to ensure that 19,000 examples were randomly selected for comparison analysis, the random filter included in the Weka tool was used. According to the results, SVM was able to obtain an accuracy of 93.95% while NB was able to obtain an accuracy of only 56.54%. SNMP-MIB dataset was used in [9] for an analysis of the basic methods that can be used to identify DDoS attacks through network traffic and the classification method they used was Random Forest. Results of a digital experiment showed that early detection is a viable option when the average assault represents 15-20% of typical traffic.

Deep learning (DL) models are becoming more and more important, and they are becoming a hot topic for research. A range of deep neural networks are included in deep learning approaches, which can be used to improve IDS effectiveness [10]. When it comes to fitting and generalization, deep learning models outperform shallow machine learning models. Moreover, deep learning techniques offer the noteworthy benefit of not requiring feature engineering or domain knowledge, setting them apart from shallow machine learning models. However, the time-consuming nature of DL models sometimes surpasses the temporal limitations required for IDS. A Deep Neural Network (DNN) with seven hidden layers and a feed-forward and back-propagation architecture was suggested in [11] for the classification of natural attacks and flows. Through traffic rerouting and anomaly monitoring, the technique reduced the impact of distributed denial of service (DDoS) attacks at the application layer. Additionally, even in the event that a new malicious pattern is employed, it can recognize the actions of a malicious packet. In [12] a convolutional neural network (CNN) was used to efficiently learn IoT features using 28050 learning parameters, and then a Long Short-Term Memory (LSTM)-based classifier was used to do the actual classifications. Pilot testing revealed that the model is effective with a success rate of up to 96%. The results showed the effectiveness of these methods in detecting intrusions. Or anomalies in network traffic, some of which achieve accuracy levels, which is critical for network security applications. In this research, we will try to reach better results and confront the challenges and weaknesses of previous experiments.

## III. METHODOLOGY

In order to create a sophisticated penetration testing system for wireless networks, security holes in the system that an attacker could exploit must be found. Vulnerability detection may be made much more accurate, flexible, and scalable by using CNNs. Figure 1 shows the several essential steps in the suggested technique for developing an AI-driven penetration testing system for wireless networks. This methodical methodology guarantees the creation of an efficient model that can accurately identify security flaws.

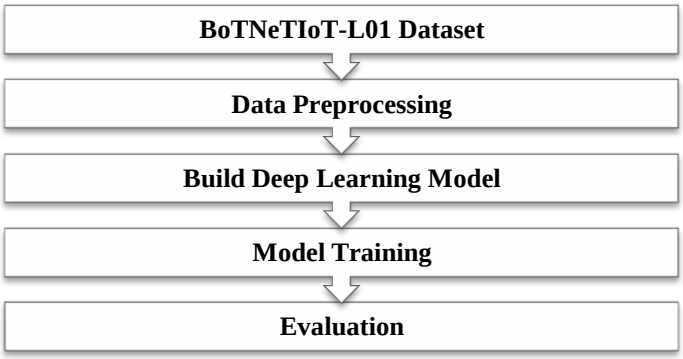


**Figure 1: Proposed methodology for create wireless network penetration testing system**

The suggested methodology makes use of the BoTNeTIoT-L01 dataset to present a thorough and efficient strategy for improving cybersecurity. Data preprocessing, model construction, training, and evaluation are the crucial elements in the methodology. With each phase, the creation of a reliable and precise CNN model that can identify and categorize IoT bot attacks with exceptional precision is ensured. The necessary packages have been imported into the Python environment such as numpy, which is an important library for learning numerical operations because it provides support for matrices and mathematical functions, as well as the Scikit-Learn library for machine learning models and Pandas for handling and analyzing data. Tensorflow is a machine learning framework developed by Google for building and training deep learning models, such as neural networks, that need a large amount of computing power and space.

### 3.1 BoTNeTIoT-L01 Dataset

BoTNeTIoT-L01 is the dataset that was used. More than 7 million data points are included in it, encompassing both normal behavior and two different IoT botnet attack types: Mirai and Gafgyt. “Mirai” is designed to infect IoT devices. It is capable of performing a variety of malicious operations, including DDoS attacks. “Gafgyt” (also known as “Bashlite”) is a form of malware that primarily targets IoT devices. Because it ensures the model's robust generalization to real-world circumstances, this large dataset is crucial for deep learning model validation and training. The dataset includes Wireshark captures of nine distinct IoT device traffic streams on a centrally switched local network which contains statistically developed features like mean, radius, size, variation, and correlation coefficient. As shown in Figure 2,

 

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 5, pp 232-237, May-2024

https://doi.org/10.47001/IRJIET/2024.805034

the data set consists of 7,062,606 records and 27 columns (features).

| | MI_dir_L0.1_weight | MI_dir_L0.1_mean | MI_dir_L0.1_variance | H_L0.1_weight | H_L0.1_mean | H_L0.1_variance |
|---|---|---|---|---|---|---|
| 0 | 1.000000 | 98.000000 | 0.000000e+00 | 1.000000 | 98.000000 | 0.000000e+00 |
| 1 | 1.931640 | 98.000000 | 1.818989e-12 | 1.931640 | 98.000000 | 1.818989e-12 |
| 2 | 2.904273 | 86.981750 | 2.311822e+02 | 2.904273 | 86.981750 | 2.311822e+02 |
| 3 | 3.902546 | 83.655268 | 2.040614e+02 | 3.902546 | 83.655268 | 2.040614e+02 |
| 4 | 4.902545 | 81.685828 | 1.775746e+02 | 4.902545 | 81.685828 | 1.775746e+02 |
| ... | ... | ... | ... | ... | ... | ... |
| 7062601 | 2.937269 | 217.763487 | 1.770682e+04 | 2.937269 | 217.763487 | 1.770682e+04 |
| 7062602 | 1.730254 | 282.630543 | 1.054589e+04 | 1.730254 | 282.630543 | 1.054589e+04 |
| 7062603 | 2.730251 | 299.980395 | 7.204117e+03 | 2.730251 | 299.980395 | 7.204117e+03 |
| 7062604 | 2.882414 | 216.723647 | 1.775308e+04 | 2.882414 | 216.723647 | 1.775308e+04 |
| 7062605 | 2.032574 | 154.377267 | 1.303249e+04 | 2.032574 | 154.377267 | 1.303249e+04 |

7062606 rows × 27 columns

**Figure 2: Samples from dataset**

### 3.2 Data Preprocessing

High-quality data input is ensured by cleaning, normalizing, and coding the dataset, which makes it easier for the deep learning model to be trained successfully. Key steps that the dataset was well prepared for model training:

- Cleaning: 621,659 duplicate rows were deleted. Table 1 shows the number of values for each category in the attack (target) column.

**Table 1: Target column distribution**

| Class | Counts |
|---|---|
| **mirai** | 3668402 |
| **gafgyt** | 2259045 |
| **Normal** | 513500 |

- Normalize Columns: MinMaxScaler is a data preprocessing technique used to scale numerical features to a specific range, most commonly between 0 and 1. It was used because it is particularly suitable for machine learning algorithms that deal with neural networks. To create a DataFrame with both the automatically encoded "Device_Name" columns and the regular numeric columns, the "Device_Name" column is encoded and linked to the DataFrame. 32 columns and 6,440,947 records are now part of the collection.
- One-hot Encoding: The process of converting categorical data into a binary value is referred to as hot encoding. A fast encoding was applied to the “attack” column containing three categorical labels to convert them to a numeric format suitable for machine learning. Figure 3 shows the target column and its three classifications after applying encryption to it.

| | Normal | gafgyt | mirai |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 |
| ... | ... | ... | ... |
| 7062071 | 1 | 0 | 0 |
| 7062072 | 1 | 0 | 0 |
| 7062073 | 1 | 0 | 0 |
| 7062074 | 1 | 0 | 0 |
| 7062075 | 1 | 0 | 0 |

6440947 rows × 3 columns

**Figure 3: Attack columns after one-hot encoding**

- Data partitioning: The data set is divided into training, validation, and test sets as shown in Table 2. The data is now segmented and ready to train and evaluate machine learning models. The set for training will be utilized for model training, the set for verification can be employed for model refinement and early termination, and the test set will be utilized to assess the efficiency of the ultimate model.

**Table 2: Data partitioning**

| | Number of Samples | Number of Features | Labels Encoded |
|---|---|---|---|
| **Training Set** | 4122205 | 32 | 3 |
| **Validation Set** | 1030552 | 32 | 3 |
| **Testing Set** | 1288190 | 32 | 3 |

### 3.3 Build Deep Learning Model

The use of convolutional neural networks, or CNNs, is commonly associated with computer vision applications such as image classification and object detection. However, it is also adaptable and applicable in other areas, such as vulnerability detection in cybersecurity and network security. We are able to teach CNNs how to recognize known malware signatures and detect strange behavior in software. CNN is a specialized deep learning model designed to process inputs organized into networks. The basic architecture of a CNN contains multiple main components and layers that cooperate to extract distinct features from input data and make predictions. This step involves constructing a CNN model using the Keras library. The model architecture is shown in Figure 4.

 

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 5, pp 232-237, May-2024

https://doi.org/10.47001/IRJIET/2024.805034

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 31, 16)            48

max_pooling1d (MaxPooling1   (None, 15, 16)            0
D)

conv1d_1 (Conv1D)            (None, 14, 32)            1056

max_pooling1d_1 (MaxPoolin   (None, 7, 32)             0
g1D)

flatten (Flatten)            (None, 224)               0

dense (Dense)                (None, 64)                14400

dense_1 (Dense)              (None, 3)                 195

=================================================================
```

**Figure 4: Proposed sequential model layers**

- Conv1D Layer: A 1D convolutional layer with 16 filters and ReLU activation function.
- MaxPooling1D Layer: Reducing the dimensions by half to down-sample the feature maps.
- Conv1D Layer: Another convolutional layer with 32 filters.
- MaxPooling1D Layer: Further down-sampling the feature maps.
- Flattening Layer: Converting the 2D output of the convolutional layers into a 1D vector.
- Dense Layer: A fully connected layer with 64 units and ReLU activation.
- Output Layer: A dense layer with 3 units and a softmax activation function to produce probability distributions for multi-class classification.

### 3.4 Model Training

The model is compiled and trained using the following configurations:

- Loss Function: "categorical_crossentropy" is used to measure the deviation between predicted and actual target values.
- Optimizer: The Adam optimizer adjusts the learning rates for each parameter individually, promoting efficient convergence.
- Metrics: Accuracy is used to track the success of the model during training and evaluation.

The model accuracy over 25 epochs for the training and validation datasets is displayed in Figure 5. The training and validation accuracies are relatively low at the start (epoch 0), hovering around 0.6. When the model first begins to learn from the data, this is normal. The accuracy of the training and validation datasets continuously improves as the number of epochs rises. This shows that the model generalizes well to the validation data and learns from the training set of data. Training and validation accuracy settle around 1.0 by epoch 25, suggesting that the model has attained good accuracy on both datasets.

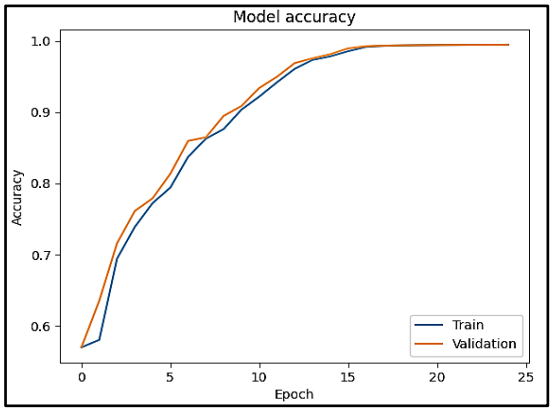


**Figure 5: Training and validation accuracy vs epochs**

The CNN model has been effectively trained, as evidenced by the accuracy trends in the graph, which show that training and validation accuracy have increased throughout the course of the epochs. The ultimate high accuracy scores demonstrate how well the model works to find weaknesses in wireless networks. The model's practical application in cybersecurity settings is highlighted by the close match between training and validation accuracy, which also suggests that the model generalizes effectively and is robust against overfitting.

### 3.5 Evaluation

The evaluation metrics presented in Figure 6 demonstrate the remarkable performance of the CNN model, which achieves excellent scores for accuracy, F1-score, precision, and recall across all categories. The confusion matrix indicates a relatively low percentage of misclassifications, which improves the model's robustness and dependability. The weighted and macro averages are nearly aligned, suggesting that the model works well in all classes, regardless of size.

```
Accuracy =  99.4693 %
--------
Precision Macro average =  98.9148 %
Precision weighted average =  99.473 %
--------
Recall Macro average =  99.4568 %
Recall weighted average =  99.4693 %
--------
F1_Score Macro average =  99.1828 %
F1_Score weighted average =  99.4703 %
--------
Confusion_Matrix :
[[102257    248    307]
 [  2036 449404    713]
 [   460   3072 729693]]
```

**Figure 6: Evaluation results**

 

ISSN (online): 2581-3048
Volume 8, Issue 5, pp 232-237, May-2024
https://doi.org/10.47001/IRJIET/2024.805034

Table 3 presents the enhanced efficacy of the suggested CNN model over earlier research, demonstrating its superior accuracy in identifying and categorizing IoT botnet assaults within wireless networks. This denoising autoencoder serves as the basis for the deep learning model developed by Abusitta et al. [13]. The goal of the model is to gain flexible characteristics of the diverse IoT environment. For the purpose of evaluating the model, the BoTNeTIoT-L01 dataset was used and the accuracy reached 94.9% on average, regardless of the number of hidden nodes, which ranged from 200 to 1000 depending on the case. Dwaibedi et al. [14] conducted an analysis on three available datasets including BoTNeTIoT-L01 using different machine learning models.

**Table 3: Comparison with previous studies in the term of accuracy**

| Paper | Method | |
|---|---|---|
| **Abusitta et al. [13]** | Autoencoder | Accuracy: 94.9% |
| **Dwibedi et al. [14]** | Random Forest<br>SVM<br>Deep learning<br>XGBoost | Best accuracy: 98.4% |
| **Proposed CNN Model** | CNN | Accuracy: 99.4693% |

When it comes to tackling the difficulties of identifying complex malware, finding fresh vulnerabilities, and responding to changing cyberthreats, the AI-driven method offers notable benefits over conventional security measures. The scalability of the system enables it to manage the growing volume and intricacy of contemporary IT settings, encompassing cloud-based resources and IoT devices. With a 99.4693% accuracy rate, a very low false positive rate, and a stellar F1 score, the CNN model created using this method performs admirably.

## IV. CONCLUSION

The widespread adoption of wireless networks and Internet of Things (IoT) devices has given rise to intricate security issues, such as advanced malware attacks. Because traditional security measures rely on signature-based techniques, they frequently fail to effectively detect these attacks. This effort aims to improve penetration testing methods by utilizing artificial intelligence, more specifically machine learning and deep learning, to better uncover security flaws in wireless networks. With the help of the BoTNeTIoT-L01 dataset, which contains more than 7 million records of both regular traffic and IoT botnet attacks (Mirai and Gafgyt), we used the Keras library to create a Convolutional Neural Network (CNN) model. Multiple Conv1D and MaxPooling1D layers, a flattening layer, and dense layers with ReLU and softmax activation functions are all part of the model design. The model underwent thorough preprocessing, which included cleaning and normalizing the data, before being tested, verified, and trained on several dataset subsets. With very few false positives and false negatives, the CNN model performed exceptionally well, achieving an astounding accuracy of 99.46%. This study highlights how AI-driven systems can detect and respond to novel cyberthreats, providing scalable and effective defenses for intricate IT systems. Future work might concentrate on putting continuous learning processes into place, updating the model with fresh information to keep it safe from new threats, and testing and fine-tuning the model in real-world situations to assess how well it works.

## REFERENCES

[1] A. A. R. Melvin, G. J. W. Kathrine, S. S. Ilango, S. Vimal, S. Rho, N. N. Xiong, and Y. Nam, "Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, 2022.

[2] D. C. Stockman, "EDUCaTION MaTTERS," *Pulse-Chartered Institute for Information Security*, 2022.

[3] Ö. A.Aslan, and R. Samet, "A comprehensive review on malware detection approaches," *IEEE access*, vol. 8, pp. 6249-6271, 2020.

[4] M. Stampar, and K. Fertalj, "Artificial intelligence in network intrusion detection," *In 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1318-1323, May. 2015.

[5] M. H. Huang, and R. T. Rust, "Artificial intelligence in service," *Journal of service research*, vol. 21, no. 2, pp. 155-172, 2018.

[6] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *Ieee Access*, vol. 6, pp. 52843-52856, 2018.

[7] R. M. Mohammad, and M. K. Alsmadi, "Intrusion detection using Highest Wins feature selection algorithm,"*Neural Computing and Applications*,vol. 33, pp. 9805-9816, 2021.

[8] A. Halimaa, and K. Sundarakantham, "Machine learning based intrusion detection system," *In 2019 3rd International conference on trends in electronics and informatics (ICOEI)*, pp. 916-920, 2019.

[9] T. Radivilova, L. Kirichenko, D. Ageiev, and V. Bulakh, "Classification methods of machine learning to detect DDoS attacks," *In 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, pp. 207-210, 2019.

 

ISSN (online): 2581-3048
Volume 8, Issue 5, pp 232-237, May-2024
https://doi.org/10.47001/IRJIET/2024.805034

[10] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *applied sciences*, vol. 9, no. 20, 2019.

[11] M. Asad, M.Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, "Deepdetect: detection of distributed denial of service attacks using deep learning," *The Computer Journal*, vol. 63, no. 7, pp. 983-994, 2022.

[12] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model," *Computer Communications*, vol. 176, pp. 146-154, 2021.

[13] A. Abusitta, G. H. de Carvalho, O. A. Wahab, T. Halabi, B. C. Fung, and S. Al Mamoori, "Deep learning-enabled anomaly detection for IoT systems," *Internet of Things*, vol. 21, 2023.

[14] S. Dwibedi, M. Pujari, and W. Sun, "A comparative study on contemporary intrusion detection datasets for machine learning research," *In 2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 1-6, 2020.

**Citation of this Article:**

Mustafa Salim Mohammed Al-Saudi, Kassem Hamze, "Artificial Intelligence-Driven Penetration Testing for Wireless Networks: Enhancing Security Vulnerability Detection Using CNN Models", Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 8, Issue 5, pp 232-237, May 2024. Article DOI https://doi.org/10.47001/IRJIET/2024.805034

*******

 