Universal Library of Engineering Technology Research Article

ISSN: 3064-996X

Open Access | PP: 08-16

DOI: https://doi.org/10.70315/uloap.ulete.2023.002

# Enhancing Cybersecurity Architectures with Artificial Intelligence (AI): A Framework for Automated Threat Intelligence Detection System

**Anand Polamarasetti[1], Rahul Vadisetty[2], Vasu Velaga[3], KishanKumar Routhu[4], Gangadhar Sadaram[5], Suneel Babu Boppana[6], Srikanth Reddy Vangala[7]**

[1]MCA, Andhra University.
[2]Wayne State University, Master of Science.
[3]Cintas Corporation, SAP Functional Analyst.
[4]ADP, Openstack Architect.
[5]Bank of America, VP DevOps/ OpenShift Admin Engineer.
[6]iSite Technologies, Project Manager.
[7]University of Bridgeport, Computer Science Dept.

## Abstract

*In order to execute cyber-security, intrusion detection systems (IDS) are developed to identify threats and irregularities in computer networks. An efficient data-driven intrusion detection system has been developed as a result of the use of artificial intelligence, particularly machine learning techniques. The proposed security model utilizes BGOTSVM to develop IDS systems starting from a security feature ranking process followed by model development using crucial features. The increasing sophistication of cyber threats necessitates robust and intelligent threat detection systems. This study uses the UNSW-NB15 dataset to demonstrate a Convolutional Neural Network (CNN)-based approach for financial fraud detection. To improve model performance, the suggested methodology incorporates data preparation techniques such as feature selection, one-hot encoding, and managing missing values. The CNN model, optimized through hyperparameter tuning, is compared against traditional machine learning (ML) models, including AdaBoost and naïve biased. Experimental findings show that CNN performs better than any baseline model, reaching the maximum accuracy (93.40%), precision (94.63%), recall (93.40%), and F1-score (92.81%). Performance evaluation metrics, classification reports, and confusion matrices further validate the CNN model's ability to identify fraudulent activity. Comparative analysis indicates that deep learning techniques, particularly CNN, offer superior threat detection capabilities by effectively identifying complex trends in communication over network information.*

**Keywords:** *Cybersecurity, Threat Detection, Network Intrusion Detection, UNSW-NB15 Dataset, Machine Learning (ML), Deep Learning (DL), Feature Engineering, Hyperparameter Tuning, Convolutional Neural Network (CNN).*

## INTRODUCTION

It makes sense that only intelligent code will provide defense against clever cyber-attacks, and recent events have shown that ransomware and cyberweapons are becoming more and more sophisticated [1]. Techniques and technological tools for cyberattacks are in existence to guard prevent harm, illegal use, alteration, and manipulation of communications and information networks and the data they hold [2]. The challenge is made more challenging by the rapid advancement of technology and the rapidly evolving nature of cyber threats. In response to this unusual challenge [3], Security teams may now more effectively reduce risks and enhance reliability with the use of AI-based cybersecurity technologies [4]. Examining the available research on using the complexity of AI and cybercrime necessitates a uniform and standardized taxonomy for cybersecurity [5]. This standardized taxonomy will help professionals and researchers better agree on the technical processes and services that require AI development to provide successful protection [6].

The purpose of an intrusion detection system (IDS) is to locate and prevent unauthorized access to any network or personal computer. It may be either hardware or applications. IDS is in two types: host-based, integrated on a host device and checks process and user activity on the local machine to detect intrusion or network-based [7]. The most commonly used IDS is established over a network and works within a network system in a distributed manner to check traffic flow for intrusions. IDS divides the Aegean Wi-Fi Intrusion Dataset (AWID) into four categories, which are the most often used: Normal, Flooding, Injection, and Impersonation, according to their behavior on the wireless network [8]. IDS attack detection can be distributed into four types: Anomaly based IDS [9], suitable for unknown attacks detection that triggers abnormal behavior; misuse-based or signature IDS

[10], ideal for detection of known attacks verified based on the predefined signature, specification-based IDS detects abnormality for network components like routing tables, nodes and protocols by a set of rules and thresholds [11].

A key element of protecting individuals and organizations is the identification of threats. Advances in artificial intelligence technologies have made it simpler to identify and eliminate threats in real time [12]. AI-based threat detection solutions enable security systems to identify risks and threats more quickly, precisely, and effectively. AI-based threat detection systems examine enormous volumes of data using algorithms and machine learning techniques to find trends that could point to potential dangers [13]. AI algorithms may be trained using a variety of data sources, including Facebook and Twitter feeds, network traffic, and photographic footage, to detect and alert security staff to possible threats or breaches of confidentiality [14].

### Significance and Contribution of Paper

Threat detection system by leveraging a CNN on the UNSW-NB15 dataset, demonstrating its superiority over traditional ML models. This study is important because it can automatically identify geographical and temporal differences in network traffic data, increasing the accuracy of threat detection. This work's primary accomplishments include:

- The implementation of CNN enables efficient and automated feature extraction, reducing reliance on manual feature engineering while improving classification performance.
- Rigorous data preparation, including dealing with values that are unavailable, one-hot encoding, and feature selection using the XGBoost classifier, ensures high-quality input data for model training.
- The CNN model is working to optimize its performance by fine-tuning crucial hyperparameters using a self-adaptive differential evolution (SADE) technique.
- The proposed CNN model is benchmarked against DT, RF, SVM, and ANN, showcasing significant improvements in accuracy (93.40%), precision (94.63%), and F1-score (92.81%).
- The model's strong functionality and flexibility in large-scale fraud detection tasks highlight its potential deployment in real-world financial security systems.

### Novality and Justification

The novelty of this research lies in leveraging a CNN for financial fraud detection using the UNSW-NB15 dataset, demonstrating its superior performance over traditional ML models. Unlike conventional approaches, such as DT and RF, which rely on manually crafted features, CNN autonomously extracts spatial and temporal patterns, enhancing fraud detection accuracy. The study further optimizes CNN through hyperparameter tuning and feature engineering, resulting in a high accuracy of 93.40% and an F1-score of 92.81%. This justifies CNN's effectiveness in handling intricate fraud structures, which makes it a viable option for practical financial protection uses.

### Structure of the Paper

The paper is organized as follows: Section II dowries applicable work ML-Enhanced Financial Fraud Detection. Section III particulars the measures, methodology and materials used. Section IV presents the investigational conclusions and result analysis, and discussion of the proposed system. Section V provides the conclusion and upcoming tasks.

## LITERATURE REVIEW

This literature review section covers several methods of intrusion detection that use artificial immune systems, data mining, and machine learning. It highlights enhanced cybersecurity threat detection and response efficacy in dynamic network environments by better categorization, data balancing, and accuracy methodologies across several datasets.

Nayak, Nadig and Ramamurthy (2019) present a method for grouping harmful URLs using ML. Concentrate on a collection of URLs collected via a framework for threat intelligence feeds. They categorize dangerous URLs retrieved from open-source threat intelligence feeds using a k-means clustering technique. Their method yielded clusters with a silhouette coefficient of 0.383 for a dataset with more than 11,000 harmful URLs. Finally, to determine the proportion of harmful terms in a given URL, it use a probabilistic scoring model. their technology effectively detects more than 80% of the URLs in a test dataset as malicious after examining more than 72,000 dangerous keywords [15].

Suliman et al. (2018) this artificial immune system (AIS) is the suggested instrument for detecting intrusions in computer networks. IDS implement a classification method to group various connection features together. The classification system enables IDS to separate between legitimate network traffic and attacks. The researchers analyzed connection data from the KDD Cup 99 competition to identify their classification type. The implemented method demonstrates its effectiveness for detecting attack connections through successful identification results [16].

Gupta et al. (2016) a number of data mining algorithms that include LR and K-Means Clustering function as the basis for automated rule generation to classify network activities. The NIDS operates as a software application that tracks network and system activities for detecting unauthorized device access and dangerous behavior. The research includes an evaluation of different techniques that detect intrusion activities. The analysis requires the NSL-KDD dataset, which provides attack patterns. Several threats emerge from network attacks that happen in the Internet environment [17].

Pal and Parashar (2014) the main objective of this paper involves developing an IDS through genetic algorithm modifications for network intrusion detection. The application of Information gain has enabled us to perform attribute subset reduction. The training process and complexity became significantly lower. Maintaining information integrity relies heavily on the security requirement known as intrusion detection. The model validation occurred by employing the KDD99 dataset. Observed evidence in the results indicates that the system detects intrusions effectively while producing few incorrect alerts [18].

Ahsan, Gomes and Denton (2018) investigates the difference in prediction accuracy between balanced and imbalanced datasets through the implementation of SMOTE data balancing. SMOTE operates to balance all data points in the dataset. Three models, namely XGBoost, RF and SVM, performed analysis on the phishing dataset. The results display substantial enhancement in accuracy when SMOTE techniques are used. XGBoost shows the most drastic increase in accuracy, from 89.87% to 97.17%, when SMOTE is used as an effective indicator for phishing data monitoring. The criminal behind phishing steals user personal information through system malware and virus attacks [19].

Abdulhammed et al. (2018) explores multi-class classification on the Aegean Wi-Fi Intrusion Dataset (AWID) using the IEEE 802.11 MAC Layer attacks as different classification categories. The proposed work reached 99.64% precision using RF with supply test and 99.99% precision through the use of RF and J48 with 10-fold cross-validation. The efficiency of an IDS constructed with machine learning techniques depends on both the classifier model and the characteristics that are picked. Marcus's learning techniques in cybersecurity identify and predict security risks before they become significant problems [20].

Table I below shows the literature review summary of various studies, approaches, datasets, main conclusions, limits, and directions for further research.

**Table 1.** Literature review summary of financial fraud detection and classification using machine learning

| Reference | Methodology | Dataset | Key Findings | Limitations & Future Work |
|---|---|---|---|---|
| Nayak, Nadig and Ramamurthy (2019) [15] | Threat intelligence K-means clustering; Probabilistic scoring | 11,000+malicious URLs; 72,000 keywords | Clustering achieved 0.383 silhouette score; Model detected 80% malicious URLs | Needs real-time analysis; explore advanced models; test on benign URLs |
| Suliman et al. (2018) [16] | AIS for Intrusion Detection | KDD Cup 99 | AIS effectively detects intrusion by grouping different connection features using classification methods. | Further refinement needed to adapt to modern attack patterns and real-time scenarios. |
| Gupta et al. (2016) [17] | Data Mining Techniques (K-Means Clustering, Linear Regression) for Network Intrusion Detection. | NSL-KDD | Putting ML into practice algorithms improves classification accuracy for malicious network activity. | Needs optimization for real-world deployment and handling dynamic threats. |
| Pal and Parashar (2014)[18] | Genetic Algorithm-Based Intrusion Detection | KDD'99 | Attribute subset reduction via Information Gain improves training efficiency and detection accuracy. | Requires further tuning to reduce false positive rates. |
| Ahsan, Gomes and Denton (2018)[19] | XGBoost, SVM with SMOTE and Random Forest for Phishing Attack Identification | Phishing Dataset | SMOTE improves detection accuracy significantly, especially for XGBoost (89.87% to 97.17%). | More generalization tests are needed for real-world phishing cases. |
| Abdulhammed et al. (2018)[20] | IEEE 802.11 MAC Layer Attacks: Multi-Class Categorization | Dataset on Aegean Wi-Fi Intrusions (AWID) | Random Forest achieved 99.64% accuracy, whereas 10-fold cross-validation yielded 99.99% accuracy. | Future work on optimizing detection speed and model interpretability. |

## METHODOLOGY

The proposed methodology for threat detection system, as illustrated in Figure 1, provides a Flowchart of the threat detection system, which begins with acquiring the UNSW-NB15 Dataset, which contains labeled transaction data. The methodology for this research involves utilizing the UNSW-Network intrusion detection using the NB15 dataset. Initially, data preprocessing is conducted, including handling missing values, checking for null values, and applying one-hot encoding. To improve data representation, feature engineering approaches like standard expanding are used. After that, the dataset is divided into two sets: 70% for training and 25% for

testing. Numerous models of categorization, including CNNs, are implemented for intrusion detection. Hyperparameter tuning is performed to optimize model performance. Finally, Performance criteria, including accuracy, precision, recall, loss, and the F1-score, are used to assess the models, and the outcomes are analysed appropriately.

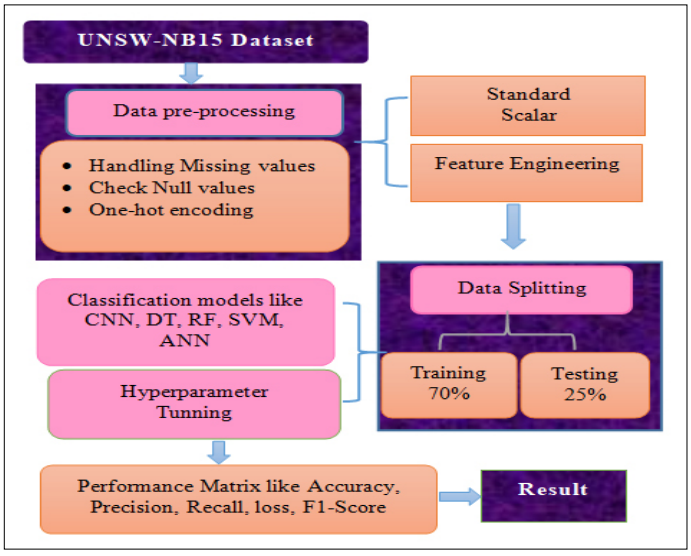


**Fig 1.** Flowchart Diagram of Threat Detection using UNSW-NB15 Dataset

A quick explanation of the flowchart's subsequent phases is provided below:

## Data Collection

The UNSW-NB15 network intrusion detection dataset was created in 2015 by the Australian Centre for Cyber Security's (ACCS) Cyber Range Lab. The raw traffic collection took place from two simulation periods totaling 15 hours during January 22 and February 17 of 2015 for a cumulative dataset size of approximately 100 GB. The dataset includes nine distinct attack groups: worms, reconnaissance, shellcode, analysis, backdoors, DoS, exploits, fizzers, and generic [21].

## Data Analysis and Visualization

The selected dataset, which consists of multiple connections being labeled between attack and normal network activities, suits supervised learning for this project, although its 47 features make it suitable. The dataset consists of two fundamental fields to show if a traffic flow constitutes an attack along with its corresponding attack classification.

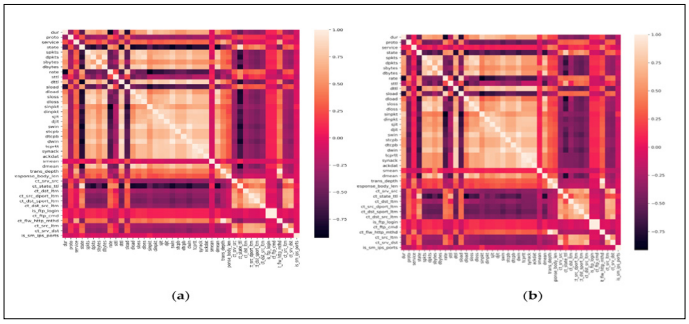


**Fig 2.** Correlation matrix of UNSW- NB15 Dataset

Figure 2 illustrates the image and displays two correlation heatmaps labeled as (a) and (b), representing the relationships among various variables. The heatmaps utilize a color gradient, where dark shades indicate strong negative associations, but stronger positive correlations are represented by brighter hues. Both heatmaps exhibit a structured pattern with a prominent diagonal, signifying high self-correlation among variables. The variable names on the x- and y-axes enable a thorough examination of feature relationships and interactions. The comparison between the two heatmaps suggests variations in correlation structures, potentially due to differences in datasets, feature selection, or preprocessing methods.

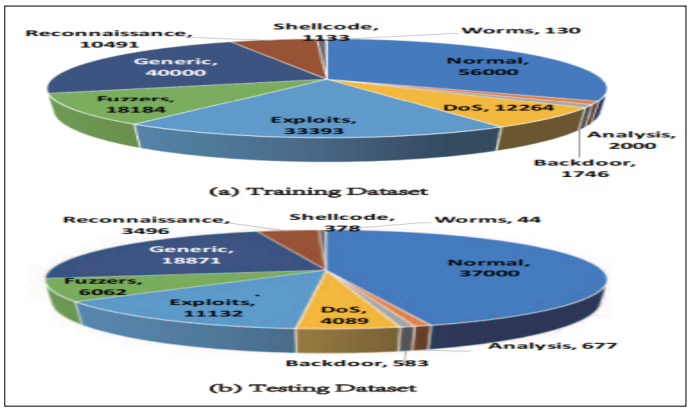


**Fig 3.** The Data Distribution of the UNSW-NB15 Dataset

Figure 3 includes two pie charts that show how different forms of attack are distributed throughout the UNSW-NB15 dataset. The first chart (a) represents the training dataset, which consists of normal (56,000) and various attack types, including Generic (40,000), Exploits (33,393), DoS (12,264), and others. The testing dataset is shown in the second chart (b), which includes normal (37,000) and attack kinds such DoS (4,089), Generic (18,871), and Exploits (11,132). Both charts visually highlight the proportion of each category, demonstrating the variety of datasets for malware detection model evaluation and training.

## Data Preprocessing

The critical first step to create dependable detection systems involves processing data especially when multiple models need comparison. For model evaluation purposes, it is vital to ensure data consistency because this ensures that test results evaluate models exclusively without being affected by how the data has been presented. The following steps will be conducted during the pre-processing phase:

- **Handling Missing values:** Effective handling techniques, such as imputation, deletion, or predictive modeling, are essential to ensure data integrity and enhance the reliability of ML models.
- **Check Null values:** To improve the accuracy of this study, the null values for these properties—ct_flw_http_mthd, is_ftp_login, and attack_cat—have been removed. As a result, the dataset has twelve rows of information type

"state," "service," "ct_ftp_cmd," "attack_cat," "srcip," "sport," "dstip," "dsport," and "proto" are the respective objects.

- **One hot encoding:** There has been a single hot encoding applied to the proto, service, and state columns. The information being encoded included 197 characters after the use of the one-hot function.

## Standard Scalar

Standard Scaler () to transform characteristics that are continuous or quasi-continuous into constant characteristics. One way to represent standardization is as Equation (1):

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Where, z = generated value, $\mu$ = mean, and $\sigma$= standard deviation.

## Feature Engineering

An ML system receives enhanced performance through feature selection when an initial collection of attributes is used to pick a suitable subset of features. Among the 197 features present in the dataset, only some properties contribute equally to performance enhancement. The XBG Classifier feature importance method enabled the identification of essential features. As shown in Figure 4, the study conducted using this strategy revealed 55 significant traits. It retrained their RF and decision tree models using the important features.

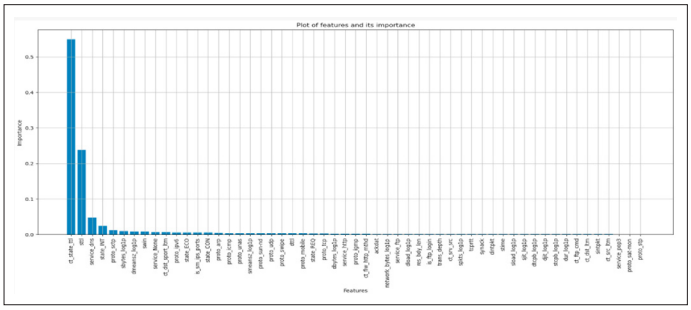


**Fig 4.** Plot of Importance of Features

The bar chart illustrates the feature importance distribution in a dataset, where the y-axis indicates each feature's relative relevance score, while the x-axis depicts different characteristics. The feature "ct_state_ttl" exhibits the highest significance, followed by "sttl" and "service_dns", while most other features contribute marginally. This visualization aids in feature selection and model optimization by identifying the most influential attributes in the dataset.

## Data Splitting

The characteristics are a subset of the dataset used to identify credit card fraud and are split by 70% for training and 25% for testing. Predictive models are built and trained using the training data, while the test data is reserved for evaluating the model's efficacy.

## Classification of Convolutional Neural Network (CNN) Model

In order to detect spatial arrangements in input data, the CNN operates as a neural network type. Weight sharing forms the base for this model since it permits one weight to serve multiple neurons across a layer. The feature map output from each layer results from performing a convolutional arithmetic operation between the layer elements and a convolutional filter matrix [22]. Consequently, in the particular instance of a 1D-CNN with a single filter, the Equation (2) [23], One way to express the feature map M∈ℝK is as a non-linear modification f of the following linear combination of the input layer's the following components:

$$M_k = f\left(\sum_y^2 \sum_{h=1}^x W_l \ .I(K-1) * s + l, h\right) \quad (2)$$

where s is the convolution's speed and K, the characteristics map's dimension, is equivalent to ⌈N+1−Ls⌉. The transposed convolutional layer exists as a second form of convolutional layer Zeiler et al. This layer functions inversely to the convolutional component for dimension restoration of the input when used concurrently [24], maintaining the information's geographical connection. Therefore, the feature map M∈ℝK comes with a dimension equal to above using the precise same approach to K=N+(L-1) *s.

## Hyperparameter Tunning

The self-adaptive differential evolution algorithm (SADE) was used to optimise the CNN model hyperparameters, including the number of filters for the first and second convolution layers (NF1 and NNH), the number of neurons in the hidden layer (NNH), the dropout rate (DR), the learning rate (LR), the batch size (BS), and the batch normalization (BN). Because of their substantial influence on model performance, these seven hyperparameters were chosen to guarantee an ideal structure while preserving computational efficiency.

## Performance Metrics

This section delves into the performance parameters that were measured during evaluation software testing that need to be evaluated. The following information regarding performance parameters used during this study will be discussed before continuation [25]. P and N stand for the total number of test instances, both positive and negative, whereas FN, FP, TP, and TN were used to determine all parameters. These parameters include the quantity of test instances classified as FN, FP, TP, and TN. TN is used to describe instances that are correctly categorized as negative, whereas TP is used to describe cases that are classed as positive but are genuinely positive [26]. The presentation of Classifiers is appraised according to f1-score, recall, accuracy, and precision.

**Accuracy:** The simplest metric to employ is accuracy. By dividing the number of correct forecasts by the total number of events, it is computed, then multiplying the result by 100. The following equation of accuracy is Equation (3):

$$Accuracy = \frac{TP+TN}{TP+Fp+TN+FN} \quad (3)$$

**Precision:** Finding the proportion of true positives among all positive forecasts, it is used to verify the system's positive

recommendations [27]. Precision is defined as Equation (4):

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

**Recall:** It speaks to the model's capacity to accurately identify transactions as fraudulent, provided that they are, in fact, fraudulent. It is used to calculate the proportion of accurately detected real positives. The recall Equation that follows is (5):

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

**F1-score:** The F1-measure (or score) syndicates two single metrics [28], It represents the precise choral mean and is described as Equation (6):

$$F1 - Score = 2\frac{(Precision*Recall)}{Precision + Recall} \quad (6)$$

**Loss:** The degree of discrepancy between the estimated value of the model and the actual value is assessed using the loss its purpose [29].

These matrices are utilized to determine the model's performance for financial fraud detection.

## RESULT ANALYSIS AND DISCUSSION

In the experimental setup, the UNSW-NB15 dataset was used to train and assess the CNN model for identifying financial fraud. The dataset was preprocessed to remove inconsistencies and normalized for optimal training performance. The CNN architecture model contains multiple convolutional layers that extract spatial features from transaction information prior to its classification stage using fully connected layers. In addition to Adam optimizer optimization, the system was trained utilizing a cross-entropy loss optimization technique. The model required adjustment of its hyperparameters together with learning rate and batch size for improved results. The model's efficacy in identifying fraudulent actions is demonstrated by the obtained accuracy of 93.40%, precision of 94.63%, and F1-score of 92.81%, which are compiled in Table II below:

**Table 2.** Results of CNN model Performance on UNSW-NB15 dataset for financial fraud detection

| Measures | Convolutional Neural Network (CNN) |
|---|---|
| Accuracy | 93.40 |
| Precision | 94.63 |
| Recall | 93.40 |
| F1-score | 92.81 |

Table II and Figure 5 presents performance evaluation of using a CNN and the UNSW-NB15 dataset. The graph displays the four primary performance indicators: accuracy, precision, recall, and F1-score. Performance is shown on the y-axis as a percentage (IN%). A 92.81% F1-score, 93.4% accuracy, 94.63% precision, and 93.4% recall are achieved by the CNN. The bars are shown as a gradation of yellow. This assessment shows how well the CNN model is for network intrusion detection.

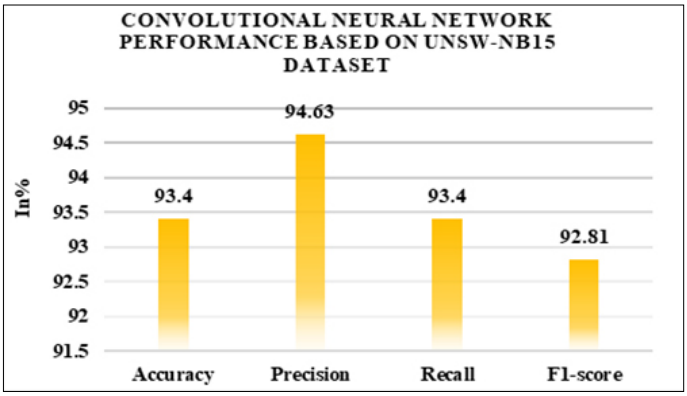


**Fig 5.** Bar Graph for CNN Model Performance

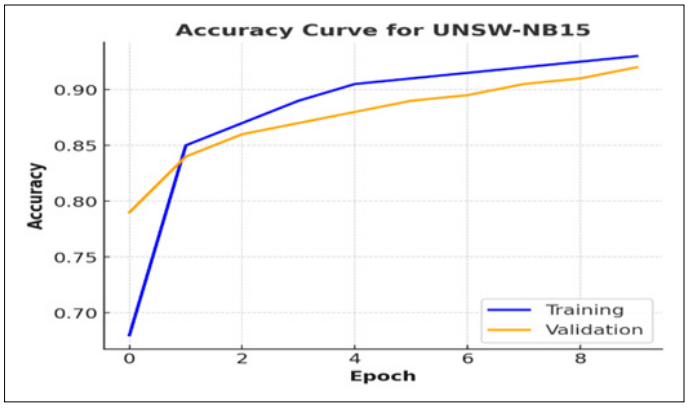


**Fig 6.** Plot for Accuracy Of CNN

Figure 6 The image depicts an accuracy curve demonstrating training and assessment accuracy throughout 10 epochs for the UNSW-NB15 dataset. The total number of epochs is displayed on the x-axis, while the accuracy levels are displayed on the y-axis. The instruction accuracy curve (blue) shows a consistent improvement as it soon gets over 0.90 after starting lower. The validation accuracy curve (orange) follows a similar trend but remains slightly below the training curve. A legend in the lower right corner differentiates the two curves. The graph includes grid lines for better readability.

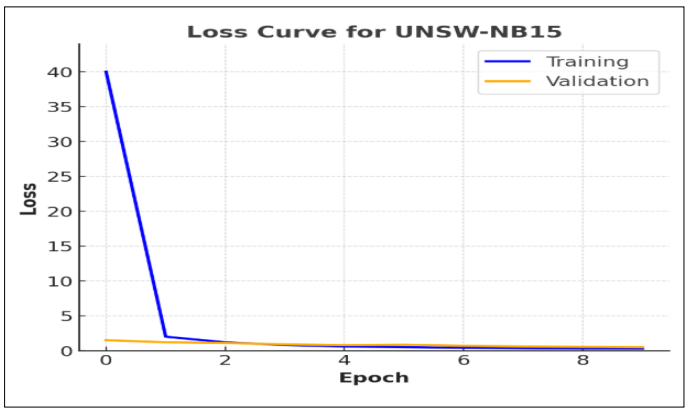


**Fig 7.** Plot of loss of CNN

Figure 7 In this graph, the loss curve for the UNSW-NB15 dataset over ten periods. The total number of epochs is displayed on the x-axis, while the loss values are displayed on the y-axis. As you can see from the blue line, training loss starts out quite high and rapidly drops to a lower number.

The orange line, which is consistent during training and quite low, represents validation loss. Both curves show a significant reduction in loss, indicating effective model learning. The distinction between training and validation loss is shown by a legend in the top right corner.

| | Accuracy | Precision | Recall | F-Score |
|---|---|---|---|---|
| Normal | 0.9835 | 0.9476 | 0.9957 | 0.9711 |
| Analysis | 0.9995 | 0.9634 | 0.9405 | 0.9518 |
| Backdoors | 0.9978 | 0.9767 | 0.5385 | 0.6942 |
| DoS | 0.9683 | 0.7778 | 0.3395 | 0.4727 |
| Exploits | 0.9444 | 0.8667 | 0.9360 | 0.9004 |
| Fuzzers | 0.9810 | 0.9810 | 0.9820 | 0.9815 |
| Generic | 0.9904 | 0.9719 | 0.7674 | 0.8576 |
| Recon | 0.9932 | 0.9780 | 0.9601 | 0.9690 |
| Shellcode | 1.0000 | 1.0000 | 0.7308 | 0.8448 |
| Worms | 0.9997 | 1.0000 | 0.8148 | 0.8980 |
| Total | 0.9340 | | | |
| Macro Avg | | 0.9463 | 0.8275 | 0.8696 |
| Weighted Avg | | 0.9321 | 0.9340 | 0.9281 |

**Fig 8.** Classification Report of CNN Model

The UNSW-NB15 dataset's arrangement performance metrics for several attack categories are shown in Figure 8 to confirm the model's effectiveness in identifying network risks. The model achieves high accuracy (93.40%) with strong precision across most classes, particularly for shellcode and worms, both attaining a precision of 1.000. However, recall values for certain attack types, such as DoS (0.3395) and backdoors (0.5385), are significantly lower, indicating difficulties in identifying all instances of these threats. The model works well overall, although it has difficulties in recognizing certain assault types, as seen by the weighted-average F1-score of 92.81% and the macro-average F1-score of 86.96%. These results indicate the need for further improvements in recall optimization to enhance threat detection performance.

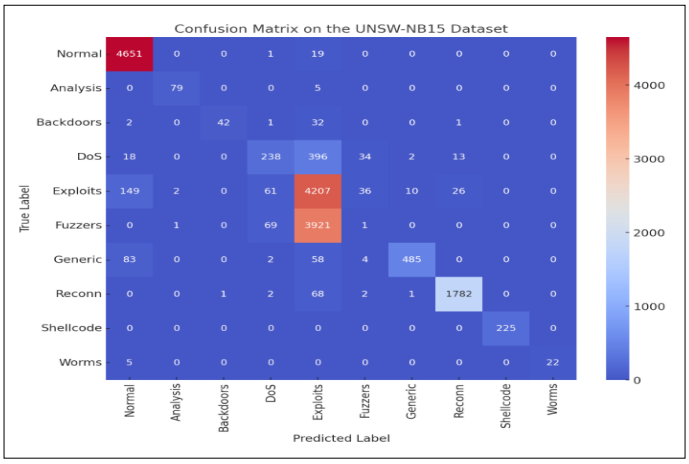


**Fig 9.** Confusion Matrix of CNN Model

The confusion matrix and arrangement performance on the UNSW-NB15 dataset are depicted in Figure 9. The highest correct predictions are for "Normal" (4,651), "Exploits" (4,207), and "Fuzzers" (3,921). Some misclassifications include 149 "Normal" instances being predicted as "Exploits" and 83 "Normal" instances as "Generic." "DoS" has 396 correct predictions but 238 misclassified as "Normal." The "Shellcode" class has 225 correct predictions with minimal misclassifications. This analysis highlights strong performance but also areas for improvement in classification accuracy.

## Comparative Analysis and Discussion

This section offers a contrast between the suggested CNN model with existing models NB [30], RF [31] and AdaBoost [32] in terms of performance on the same dataset. Table III provides examples of the following model comparisons.

**Table 3.** ml models comparison for threat detection on unsw-nb15 dataset

| Models | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| NB[30] | 74.19 | 75.50 | 92.16 | 83.11 |
| AdaBoost [32] | 86.40 | 86.74 | 93.3 | 89.94 |
| RF[31] | 75.5 | 75.5 | 75.5 | 72.4 |
| Proposed CNN | 93.40 | 94.63 | 93.4 | 92.81 |

A comparison of the several machine learning models employed for threat detection on the UNSW-NB15 dataset is shown in Table III. NB, AdaBoost, RF, and a planned CNN are among the models that are being compared. Among these, the proposed CNN model outperforms the others across all evaluation metrics, achieving the highest accuracy (93.40%), precision (94.63%), recall (93.4%), and F1-score (92.81%). AdaBoost also shows strong performance, especially in recall (93.3%) and F1-score (89.94%). In contrast, NB and RF models perform moderately, with lower overall metrics. This comparison highlights the effectiveness of deep learning techniques like CNN for detecting threats in complex network traffic data.

## CONCLUSION AND FUTURE DIRECTION

Cybersecurity detection of hazards has new opportunities to increase detection accuracy and efficiency. AI can efficiently react to increasingly complex assaults, identify possible threats quickly, and provide improved safeguarding solutions by using a range of techniques, including ML, DL, and integrated learning. The proposed CNN-based threat detection model demonstrates superior performance in identifying fraudulent activities compared to traditional models for machine learning; the UNSW-NB15 dataset yielded the greatest accuracy (93.40%), precision (94.63%), recall (93.40%), and F1-score (92.81%). The findings demonstrate how well DL can identify intricate patterns in network traffic data, making CNN a reliable choice for financial fraud detection.

Future research will concentrate on using cutting-edge DL to increase model resilience architectures such as Transformer-based models and attention mechanisms. Additionally, real-time fraud detection, adaptive feature selection, and integration with federated learning approaches will be explored to further improve system efficiency, scalability, and security in dynamic financial environments.

## REFERENCES


1. M. Semerci, A. T. Cemgil, and B. Sankur, "An intelligent cyber security system against DDoS attacks in SIP networks," *Comput. Networks*, 2018, doi: 10.1016/j.comnet.2018.02.025.
2. Y. Cherdantseva *et al.*, "A review of cyber security risk assessment methods for SCADA systems," 2016. doi: 10.1016/j.cose.2015.09.009.
3. V. Kolluri, "A Comprehensive Analysis on Explainable and Ethical Machine: Demystifying Advances in Artificial Intelligence," *Int. Res. J.*, vol. 2, no. 7, 2015.
4. P. Pranav, "Research in Computer Applications and Robotics Artificial Intelligence Education : Emotional," *Int. J. Res. Comput. Appl. Robot.*, vol. 4, no. 2, pp. 24–28, 2016.
5. Catherine Stupp, "Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case - WSJ," *Wall Str. J.*, 2019.
6. A. K. Mohemmed Sha, "Artificial Intelligence in Cyber Security_ A Survey," 2016.
7. D. D. Rao, "Multimedia based intelligent content networking for future internet," *EMS 2009 - UKSim 3rd Eur. Model. Symp. Comput. Model. Simul.*, pp. 55–59, 2009, doi: 10.1109/EMS.2009.108.
8. M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," *IEEE Trans. Inf. Forensics Secur.*, 2017, doi: 10.1109/TIFS.2017.2762828.
9. H. Li, F. Wei, and H. Hu, "Enabling dynamic network access control with anomaly-based IDS and SDN," in *SDN-NFV 2019 - Proceedings of the ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization, co-located with CODASPY 2019*, 2019. doi: 10.1145/3309194.3309199.
10. A. Gogineni, "Novel Scheduling Algorithms For Efficient Deployment Of Mapreduce Applications In Heterogeneous Computing," *Int. Res. J. Eng. Technol.*, vol. 4, no. 11, p. 6, 2017.
11. M. H. Dar and S. Z. Harahap, "Implementasi Snort Intrusion Detection System (IDS) Pada Sistem Jaringan Komputer," *J. Inform.*, vol. 6, no. 3, pp. 14–23, Sep. 2017, doi: 10.36987/informatika.v6i3.1619.
12. P. Boyer and B. Bergstrom, "Threat-detection in child development: An evolutionary perspective," 2011. doi: 10.1016/j.neubiorev.2010.08.010.
13. D. J. Stein and R. M. Nesse, "Threat detection, precautionary responses, and anxiety disorders," 2011. doi: 10.1016/j.neubiorev.2010.11.012.
14. V. Kolluri, "A Pioneering Approach To Forensic Insights: Utilization AI for Cybersecurity Incident Investigations," *Int. J. Res. Anal. Rev.*, vol. 3, no. 3, 2016.
15. S. Nayak, D. Nadig, and B. Ramamurthy, "Analyzing Malicious URLs using a Threat Intelligence System," in *International Symposium on Advanced Networks and Telecommunication Systems, ANTS*, 2019. doi: 10.1109/ANTS47819.2019.9118051.
16. S. I. Suliman, M. S. Abd Shukor, M. Kassim, R. Mohamad, and S. Shahbudin, "Network Intrusion Detection System Using Artificial Immune System (AIS)," in *2018 3rd International Conference on Computer and Communication Systems, ICCCS 2018*, 2018. doi: 10.1109/CCOMS.2018.8463274.
17. D. Gupta, S. Singhal, S. Malik, and A. Singh, "Network intrusion detection system using various data mining techniques," in *International Conference on Research Advances in Integrated Navigation Systems, RAINS 2016*, 2016. doi: 10.1109/RAINS.2016.7764418.
18. D. Pal and A. Parashar, "Improved genetic algorithm for intrusion detection system," in *Proceedings - 2014 6th International Conference on Computational Intelligence and Communication Networks, CICN 2014*, 2014. doi: 10.1109/CICN.2014.178.
19. M. Ahsan, R. Gomes, and A. Denton, "SMOTE Implementation on Phishing Data to Enhance Cybersecurity," in *IEEE International Conference on Electro Information Technology*, 2018. doi: 10.1109/EIT.2018.8500086.
20. R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Alessa, "Enhancing Wireless Intrusion Detection Using Machine Learning Classification with Reduced Attribute Sets," in *2018 14th International Wireless Communications and Mobile Computing Conference, IWCMC 2018*, 2018. doi: 10.1109/IWCMC.2018.8450479.
21. M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Multi-classification of UNSW-NB15 dataset for network anomaly detection system," *J. Theor. Appl. Inf. Technol.*, 2018, doi: 10.1007/978-981-15-5077-5_40.
22. R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018, doi: 10.1007/s13244-018-0639-9.
23. Y. H. Rajarshi Tarafdar, "Finding Majority for Integer Elements," *J. Comput. Sci. Coll.*, vol. 33, no. 5, pp. 187–191, 2018.
24. J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, 2018, doi: 10.1016/j.patcog.2017.10.013.
25. B. A. C. Groen, M. J. F. Wouters, and C. P. M. Wilderom, "Employee participation, performance metrics, and job performance: A survey study based on self-

determination theory," *Manag. Account. Res.*, 2017, doi: 10.1016/j.mar.2016.10.001.

26. N. Khare and S. Yunus Sait, "Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models," *Int. J. Pure Appl. Math.*, vol. 118, no. 20, pp. 825–838, 2018.
27. A. H. Anju, "Extreme Gradient Boosting using Squared Logistics Loss function," *Int. J. Sci. Dev. Res.*, vol. 2, no. 8, pp. 54–61, 2017.
28. H. Huang, H. Xu, X. Wang, and W. Silamu, "Maximum F1-score discriminative training criterion for automatic mispronunciation detection," *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2015, doi: 10.1109/TASLP.2015.2409733.
29. P. G. Curtis, C. M. Slay, N. L. Harris, A. Tyukavina, and M. C. Hansen, "Classifying drivers of global forest loss," *Science (80-. ).*, 2018, doi: 10.1126/science.aau3445.
30. M. Belouch, S. El Hadaj, and M. Idhammad, "Performance Evaluation of Intrusion Detection Based on Machine Learning Using Apache Spark," *Procedia Comput. Sci.*, vol. 127, pp. 1–6, 2018, doi: 10.1016/j.procs.2018.01.091.
31. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2895334.
32. M. M. Baig, M. M. Awais, and E. S. M. El-Alfy, "A multiclass cascade of artificial neural network for network intrusion detection," in *Journal of Intelligent and Fuzzy Systems*, 2017. doi: 10.3233/JIFS-169230.
33. Kuraku, D. S., Kalla, D., & Samaah, F. (2022). Navigating the link between internet user attitudes and cybersecurity awareness in the era of phishing challenges. *International Advanced Research Journal in Science, Engineering and Technology*, *9*(12).
34. Kalla, D., Smith, N., Samaah, F., & Polimetla, K. (2022). Enhancing Early Diagnosis: Machine Learning Applications in Diabetes Prediction. *Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-205. DOI: doi.org/10.47363/JAICC/2022 (1)*, *191*, 2-7.
35. Kalla, D., Kuraku, D. S., & Samaah, F. (2021). Enhancing cyber security by predicting malwares using supervised machine learning models. *International Journal of Computing and Artificial Intelligence*, *2*(2), 55-62.
36. Kalla, D., Smith, N., Samaah, F., & Polimetla, K. (2021). Facial Emotion and Sentiment Detection Using Convolutional Neural Network. *Indian Journal of Artificial Intelligence Research (INDJAIR)*, *1*(1), 1-13

**Citation:** Anand Polamarasetti, Rahul Vadisetty, et al., "Enhancing Cybersecurity Architectures with Artificial Intelligence (AI): A Framework for Automated Threat Intelligence Detection System", Universal Library of Engineering Technology, 2023; 08-16. DOI: https://doi.org/10.70315/uloap.ulete.2023.002.