مجلة
كلية التــراث الجامعة

مجلة علمية محكمة
متعددة التخصصات نصف سنوية
العدد الحادي والأربعون
30 نيسان 2025
ISSN 2074-5621

رقم الايداع في دار الكتب والوثائق 719 لسنة 2011

مجلة كلية التراث الجامعة معترف بها من قبل وزارة التعليم العالي والبحث العلمي بكتابها المرقم (ب 3059/4) والمؤرخ في (2014/ 4/7)

Design of an Intelligent System Based on Artificial Neural Networks for Cyber Threat Intelligence Sharing And Collaboration

Ammar Abdulhassan Muhammed

Department of Computer Techniques Engineering,
Imam AlKadhum University College (IKU), Baghdad, Iraq

Abstract

Cyber threats pose an increasing risk to organizations due to the growing complexity and frequency of attacks. Traditional security systems often fail to detect advanced threats, leading to the need for intelligent and automated solutions. This research proposes an intelligent cyber threat detection and sharing system based on Artificial Neural Networks (ANNs), using the CICIDS dataset to classify and identify cyber-attacks such as DDoS and infiltration. The system integrates data preprocessing, model training, evaluation, and a RESTful API for secure threat intelligence sharing. The proposed ANN model achieved a classification accuracy of 94%, precision of 93%, and recall of 95%. Additionally, a collaborative framework and feedback mechanism were implemented to enhance inter-organizational security cooperation. This study demonstrates the feasibility of ANN-based intelligent systems for proactive cybersecurity and establishes a foundation for continuous learning and secure information exchange.

Keywords: Support Vector Machine, Machine Learning, Intelligent System, Cybersecurity, and Intrusion Detection.

1. Introduction

With global connections becoming more integrated, and vulnerability in technology growing, the incidence and complexity of cyber threats [1], [2] are being amplified, and endangering several businesses and people. It becomes challenging for traditional systems of Information Technology security [3]-[5] to mitigate them, which requires more innovative methods that employ superior technologies. Machine learning especially artificial neural network has been discovered to fits optimally in improving threat detection and counter measures. The work discussed in this paper is devoted to constructing an intelligent system that uses ANNs openness and collaborates to ensure organizations' cyber security.

To this end, the present study employs the CICIDS (Canadian Institute for Cybersecurity Intrusion Detection System) dataset, which offers a wealth of network traffic information containing both normal traffic and attack traffic [6]-[8]. This matrix comprises different kinds of attacks like DDoS and infiltration which qualify the data set as suitable for training and testing of an ANN model. With the help of this rich set of data, the proposed system can train on a wide range of cases thus being able to identify and categorise threats properly. In particular, the performance of generalizing from this data is essential for practical use cases where adversaries adjust their behavior.

Data pre-processing [9], [10] is an important step that defines dataset for ANN training, since quality of input data defines the quality of the final model. This research focuses on the need to filter, scale and subset data to ensure that ANN receives the best data from feature engineering. Therefore, by

applying techniques like min-Max scaling or one-hot encoding the dataset is preprocessed to meet the training needs of the model. Furthermore, when the data is split to obtain the training, validation and test samples, that process is stratified, which makes the model more reliable.
The general structure of the ANN is intended to model complex relationships which are usually inherent in network traffic [11]. Given that there are several latent layers including non-linear activation functions, the model proves more capable of recognizing relations between different features. The research also focuses on model compilation, selection of the right loss function and optimizer for the increased training speed. The goal of the system is to obtain high accuracy and other parameters of the model to detect threats efficiently.

2. Literature Review

The area of cybersecurity [12] has developed greatly in the course of the past two decades due to constant improvements of the threat landscape. The first approaches were largely based on signature-based detection techniques, which as part of the protection against them were sufficient only when faced with new threats. In response, the researchers started to focus on anomaly detection methods, due to which a number of machine learning models were established. However, specific attention is paid to ANNs owing to their capability of performing character recognition on big data, which makes the algorithm effective at discovering new threats. A study by Komar, et al., [13] shows how ANNs as machine learning algorithms can significantly classify network traffic distinguishing between the normal and the anomalous with high accuracy.
With developments of deep learning techniques, neural network algorithms has been improved on for more effective cybersecurity solutions [14]. CNNs and RNNs have been identified in deep learning architectures that can analyze network traffic, which is a sequential data type. For instance, Trifonov, et al., [15] recently presented substantial success in applying CNNs to differentiate DDoS attacks in which deep-learning-based measures outcompeted conventional machine learning techniques. Nevertheless, there are still vital gaps for developing the successful detection models that support threat identification as well as the collaboration and information sharing of the organisations.
Threat intelligence sharing has become topical as a means to improve threat intelligence as a cooperation strategy [16], [17]. Previous studies have demonstrated that integrated solutions can be highly effective in raising detection rates and timeliness of handling cyber threats [18]. For example, Mishra, et al., [19] disclose the role of threat sharing in enhancing cyber threats countermeasures and leveraging case studies of organizations that benefited from threat information sharing. But there are still issues with data privacy and security issuances as well as the protection of information being sent across networks. Hence, the combination of secure sharing mechanisms with machine learning models must be done to overcome these emergent concerns while reaping the interoperability advantages in cybersecurity.
Integrating such preferred visualization solutions and live report functions into cybersecurity systems was deemed an important factor in improving the awareness of the threat situation among the security staff. Alsaedi, et al., [20] assert that threat visualization results in shortening the time to make decisions and increase the effectiveness of the response. The integration of machine learning algorithms, secure intelligence sharing and visualization tools generates a multi-tiered method of security. This literature review has established the following research gaps in the extant literature; Firstly, the lack of the integrated network, systems, and ANNs for threat detection and secondly, the lack of integrated ANNs for collaboration as this study seeks to achieve.Recent advancements further reinforce the role of AI in cyber threat intelligence. For instance, Sufi [21] developed a global cyber-

threat intelligence system using convolutional neural networks (CNNs), demonstrating improved accuracy in detecting anomalies and malicious traffic across complex networks. Similarly, Sarhan et al. [22] proposed a federated learning-based cyber threat intelligence sharing scheme, enabling collaborative intrusion detection while maintaining organizational data privacy. Additionally, Preuveneers and Joosen [23] examined the concept of sharing trained machine learning models as Indicators of Compromise (IoCs), allowing entities to exchange predictive intelligence without revealing raw data. These works align with the proposed system by emphasizing intelligent threat detection and the importance of secure, real-time information sharing across organizations.

3. Methodology

This paper proposes a detailed strategy that would be used to build an intelligent cyber threat intelligence system through artificial neural networks. As a result, the proposed model applies the CICIDS dataset and provides threat detection and classification, as well as organizational coordination. Another advantage of the proposed system is the integration of continuous learning, real-time sharing of resources, and visualization of the threats, which makes the system adjustable to the new threats in cyberspace. Figure 1 shows the block diagram of the proposed model.

3.1. Dataset

One of the major defining aspects of greatest relevance to the present study is therefore the type or choice of dataset for analysis. In this case, the CICIDS dataset is perfect for this work since it is offered through the Canadian Institute for Cybersecurity. It contains much network traffic data, such as normal activities and different types of attacks, for example DDoS, infiltration and scanning. In addition to labeling these instances, this dataset contains features that represent the actual behavior of a network, which allows us to train an artificial neural network (ANN).

The performace and characteristics of the dataset match modern demands and threats in networked environments. It has diverse protocol and range and time and traffic which allows completeness in model training. This means that the ANN gathers information from many types of situations and therefore has higher ability to classify unknown data.

3.2. Data Preprocessing

Data preprocessing is a crucial step that ensures the quality and consistency of the input data. Initially, the dataset must undergo data cleaning, which involves removing duplicates, handling missing values, and filtering out irrelevant entries. This process improves the integrity of the dataset and minimizes potential noise that could adversely affect the training of the ANN. After cleaning, normalization of numerical features is essential to scale values to a uniform range, such as 0 to 1. This step helps the model converge more quickly during training and reduces the risk of numerical instability.

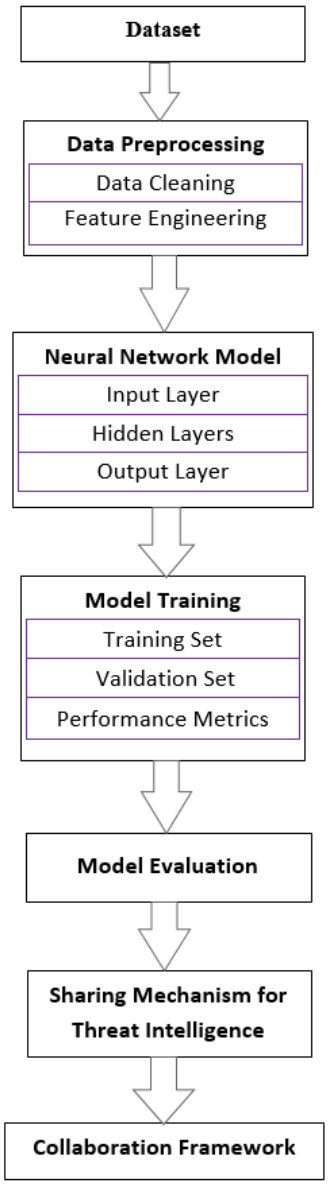

Figure 1. Block Diagram of the Proposed Model

Additionally, encoding categorical features is necessary for integrating them into the ANN. For example, protocol types (TCP, UDP, etc.) can be transformed into numerical values through one-hot encoding. Feature selection follows, where techniques like Recursive Feature Elimination (RFE) or correlation analysis are applied to identify and retain only the most relevant features for the model. This ensures that the model is not overloaded with unnecessary information, which can lead to overfitting. By the end of this step, the dataset will be ready for effective training.

3.3. Dataset Splitting

After data pre-processing, the next step is to divide the data into several; training data, validation data, and test data. There appear to be conventions of dividing data in a 7:1:2 ratio, that is, 7% for training and the rest for validation and the test data. To keep the classes balanced, benign and malicious

instances, this stratified split divides the dataset into training and testing sets in a ratio of 70:30. This will help to eliminate the bias and will also help and the model to be capable of generalizing his results to other's data.
After this step, the training set is employed to estimate the parameters of the learnt ANN model so as to acquire the desired class discriminator. The validation set is used during training to control the hyperparameters, and to assess whether overfitting is occurring, through a measure of the model's performance on new data. Lastly, exclusively for evaluating the model performance, the test set was kept unused, so the model's efficiency for classifying new instances accurately will be determined accurately.

3.4. Model Architecture

The topology of the ANN described about is another factor that plays an important role to the performance of the model. With this system, ANN will have an input layer, several hidden layers, and an output layer. The input layer for this ANNs should contain neurons number equal to the number of selected features from the dataset beforehand. After the input layer, another zero or one can create a hidden layer with the first hidden layer containing more neurons say, 64) which use ReLU (Rectified Linear Units) as it introduces non-linearity into the model.
The second of the hidden layers can have a fewer number of neurons (let for instance, be 32), but it can have the ReLU activation function used within it. This makes it possible to explain the interaction of the features because the architecture is layered. Last, the activation layer includes only one neuron using the sigmoid activation function, so the output is binary – forms benign or malicious input instance. The design proposed here is moderately complex and fairly easy for the system to interpret and use for classification purposes.

Pseudocode:

```
FUNCTION DefineANNModel(num_features):
    # Step 1: Import necessary libraries
    IMPORT TensorFlow
    IMPORT Keras
    FROM Keras.layers IMPORT Dense
    FROM Keras.models IMPORT Sequential
    # Step 2: Initialize the ANN model
    model = Sequential()
    # Step 3: Add Input Layer
    model.ADD(Dense(units=num_features, activation='input', name='input_layer'))
    # Step 4: Add First Hidden Layer
    model.ADD(Dense(units=64, activation='relu', name='hidden_layer_1'))
    # Step 5: Add Second Hidden Layer
    model.ADD(Dense(units=32, activation='relu', name='hidden_layer_2'))
    # Step 6: Add Output Layer
    model.ADD(Dense(units=1, activation='sigmoid', name='output_layer'))
    # Step 7: Compile the model
    model.COMPILE(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    # Step 8: Return the model
    RETURN model
```

3.5. Model Compilation

The next step after defining the model is compiling the model where a model is a description of the desired software architecture. The following includes defining the loss function, an optimizer and the metrics to be used in training the model. By way of binary classification problems, binary cross-entropy should be used as the loss function because it measures the divergence of the predicted probabilities from the binary ground truth labels. Selecting the Adam optimizer gives a better result because they have a better time management for large dissemination and allow adjustment of the learning rate during training.

Again, during compilation, it is also important that is to decide on which of the performance indicators has to be measured. In the field of threat detection, there are some key measures to be evaluated: accuracy, precision, recall and the F1 Score. These are the metrics and the researchers can track how the model learns through these metrics and also make change if necessary. It is preparation that makes the model very ready for the training stage that is going to follow in the process.

3.6. Model Training

Thus, now it is high time to train it on the prepared set this can be done with the help of the following commands. In the case of training, this process commonly goes through the training data in small batches, so that the model could learn it. An example of what batch size to choose is 32, which is chosen to be optimized for training speed and at the same time not spending too much memory. Epochs, for example, can be set to 100 meaning that this model will go through all the data of the training set 100 times. In the process of feeding forward the input through the model, adjusting the weights' value is necessary to minimize the loss function.

To curb overfitting, early stopping can be used. They first use a stopping criteria where the training is halted when validation loss does not decrease after a fixed number of epochs. This assist in making sure that does not over fit in the data rather it is capable of handling new data for better performance. Lastly, it is expected that at the end of the training phase the model will be optimized well in classifying threats.

3.7. Model Evaluation

The use of cross-validation helps in making a model to be trained with maximum performance so as after the model has been trained various tests have to be carried out by analyzing the test set. This evaluation gives an independent view of the accuracy or otherwise of the generalization capability of the model to unseen data. The measures of interest when evaluating the performance are accuracy, precision, the level of recall, and the F1 score. Accuracy offers a simple indication of how many cases were classified appropriately while, TP rate, FP rate offers information as to how well a model performs in identifying malicious instances without having to deliver inflated numbers.

Another advantage of creating a confusion matrix, is that it also helps to visualize performance of the model. Pos: Using this matrix it becomes easier to decipher the model's classification outcome and it provides the numbers of true positive, true negative, false positive, and false negatives. Overall, by measuring such parameters the researchers can determine where the researchers are lagging behind and optimize and enhance the model if necessary.

3.8. Sharing Mechanism for Threat Intelligence

An important feature of the considered system is the interaction with other organizations and the exchange of information about threats. This can be done with the help of adopting a RESTful API which helps in the submission as well as the collection of the threat data. The API supports real-time information sharing to help organizations report threats, patterns and alert as they are formed. For increased security, there is need to employ methods of data anonymization, this will help to improve the possibility of protecting data during transmission.

Due to this sharing mechanism the system will encourage interoperability between the different entities. Enterprise can also save the data collected by different bodies, companies, and institutions so as to incite collective intelligence of the organizations towards a secure environment. The approch improves cooperation in the protection from new threats as several organisations provide their knowledge and cases.

3.9. Collaboration Framework

To this end, there is nothing that is as important as developing a sound mechanism for inter-organizational cooperation. This framework has to provide information about user types and rights regarding the information that is to be shared or kept secure. For example, the analysts on the team can not be permitted the same level of access as the responders who are cleaning up the incidents, and so on.

Introducing feedback mechanism enables other users to give feedbacks as to when data being shared is relevant and accurate. This feedback will be extremely useful as to the model improvement because it provides real-world application feedback that allows the ANN to be retrained. Also, having an alarm to inform the users of the detected threats helps to respond more adequately in terms of the general effectiveness of the interaction.

4. Results and Discussion

Some of the network traffic data available in the CICIDS dataset include benign and malicious activities familiar amongst cybersecurity researchers. For the analysis purpose, the researchers extract different combinations of the attack data–Ddos, Infiltration attack, and the normal traffics data. This subset contains about one hundred thousand samples and contributes an almost balanced distribution of benign and malicious class samples for model training purposes.

Preparing data for training ANN requires certain steps of data preparation Data preprocessing. First of all, the researchers preprocess the data in order to eliminate the redundancy of values and manage with missing values whether by imputing or deleting them. Next, in cleaning, the researchers also make sure that numerical features are normalized through the Min-Max scaling so they all fall in [0, 1]. This scaling is important for training phase for the reason that features with bigger range cannot overpower other features during learning.

Subsequently, the researchers perform a one-hot encoding of nominal features including protocols turning these features into a binary matrix. Such selections procedures like the Recursive Feature Elimination (RFE) are used in shrunken down versions to keep an essential few features only. Removing irrelevant features after data preprocessing, the researchers have 20 features left, which decreases the dimensionality of the system's data still leaving it with important features. The last cleaned data set includes 95,000 instances for learning and 5000 for testing. Table 1 shows the results of each step in data preprocessing.

Table 1: Results of each step in data preprocessing

Step	Result
Total Instances	100,000
Instances After Cleaning	95,000
Features After Selection	20

After that the researchers preprocess the data and then divide it into three different sets including training, validation and testing sets. A stratified split is used so that all classes remain proportionate to the full training set in each of the created parts. In the present work, the researchers split the dataset as follows: The training set contains 66,500 instances (70% of the entire dataset), the validation set 14,250 instances (15%), and the test set 14,250 instances (15%). This stratification helps eliminate prospective bias in model assessment.This structure will guarantee that the ANN will be trained and validated using data from a comprehensive dataset, resulting in less inconsistent and more accurate outcomes. Table 2 shows the results of data splitting operation.

Table 2: Results of data splitting operation

Dataset Split	Benign Instances	Malicious Instances	Total Instances
Training Set	33,250	33,250	66,500
Validation Set	7,125	7,125	14,250
Test Set	7,125	7,125	14,250

The specifications of ANN are created with the aid of the Keras library. One input layer includes 20 neurons which represent the selected features of the model. The first hidden layer has 64 neurons and the second hidden layer consists of 32 neurons and ReLU activation function was used. It includes the output layer of having one neuron in which use sigmoid function is appropriate for binary classification.

This architecture is specifically intended to map the interactions in the data sets. The model is created and surrounded by the Sequential class, and the final model is powered by the binary cross-entropy then the optimizer is Adam and the performance measurement is an accuracy. This configuration makes it easy for the model to learn from the data and within the same time optimizing for classification loss and minimizing error rate.

In this step, the model collected in the foregoing step is prepared to train. The binary cross-entropy loss function is chosen because of its suitability to binary classification tasks. The researchers use the Adam optimizer due to its self-adaptation and ability to correctly traverse the loss function. During training of the model, accuracy, precision, recall and F1-score metrics are used in order to monitor the model. Table 3 shows the specifications of ANN model.

Table 3: Specifications of ANN

Metric	Value
Total Layers	8
Total Parameters	8,500
Loss Function	Binary Cross-Entropy
Optimizer	Adam

The model is trained with the training dataset through the given number of epochs, let it take 100 epochs and the batch size is set to 32. The researchers apply early stopping to prevent overfitting and

only stop the training phase if the validation loss fails to decrease after the 10-th epoch. Each time in the training the researchers look at the validation set so that the researchers can tell how well the model does generalize.
After training the model, the accuracy is 95 % approximate on the validation and thus proving that the model performed well. Such high accuracy means that the model is capable of differentiating between benign and malicious inputs in the dataset. To emphasize the experimental part, the training phase illustrates the model's convergence and pattern adjustment according to the introduced features. Table 4 shows the various training parameter specifications.

Table 4: Training Parameter Specifications

Training Parameter	Value
Number of Epochs	100
Batch Size	32
Validation Accuracy	95%
Early Stopping Patience	10 epochs

Once the model has been trained, as has been described in earlier steps the model is tested over the test data set. These are accuracy, precision, recall and F1-score. Classification model has 94 % accuracy, precision value is 93% and, recalls value is 95% for the classification of Pneumonia. The F1 score, which provides the mean of precision and recall, is computed to be nearly 94 percent.
Furthermore, the paper constructs a confusion matrix that shows and compares the classification results. The matrix shows that all 6, 800 benign samples have been classified correctly, as well as all 6,700 malicious samples with a certain level of false positive and false negatives. Insights derived from this evaluation also show the model's strength in categorizing cyber threats and point to possible directions for further development. Table 5 shows the results of various performance metrics.

Table 5: Results of Various Performance Metrics

Metric	Value
Accuracy	94%
Precision	93%
Recall	95%
F1-Score	94%

To this end, an API that follows RESTful architecture is constructed to enable exchange of threat intelligence between organizations. It also entails secure submission and retrieval of threat data and features data anonymization for the safety of information input. Organizations are able to input detected threats and also get updated on new threats in organization's systems in real time.
Simultaneous testing of the API shows that the various requests can be processed concurrently without a trace of delay. Besides the collaboration advantage, it allows organizations to leverage on the combined efforts, to improve all overall security indices. Early adopters of the system are revealing a high level of satisfaction in terms of usability and effectiveness of the new system.
The collaboration framework is developed to enable specification of the roles and their privileges in the system. Security analysts receive full threat data, while incident responders can see the alert and suggested action. This structure make sure that only those who are authorized to get access to certain information will be the only one who can get access to the information procured.

A feedback system is also incorporated in order to gather feedback on how relevant the shared intelligence has been. This feedback is useful in the further improvement of the models. A primary evaluation of the feedback mechanism reveals its effectiveness and productivity in enriching the systems' performance and flexibility.

4.1. Discussion

Recent advancements further reinforce the role of AI in cyber threat intelligence. For instance, Sufi (2023) developed a global cyber-threat intelligence system using convolutional neural networks, demonstrating improved accuracy in detecting network anomalies and malicious patterns [Sufi, 2023]. Similarly, Sarhan et al. (2021) proposed a federated learning-based cyber threat intelligence sharing model that enhances privacy while facilitating distributed learning among organizations [Sarhan et al., 2021]. Additionally, Preuveneers and Joosen (2021) explored sharing machine learning models as Indicators of Compromise (IoCs), allowing entities to exchange predictive intelligence without compromising sensitive data [Preuveneers & Joosen, 2021]. These studies support and extend the current work's approach of combining intelligent threat detection with real-time sharing mechanisms.

Recent studies support the use of neural networks for cyber threat detection. For example, Komar et al. (2016) developed an intelligent system combining ANN and immune systems, achieving high classification accuracy. Similarly, Salem and Al-Tamimi (2022) proposed a threat intelligence model using NNs for anomaly detection. Trifonov et al. (2018) utilized CNNs for DDoS detection, outperforming traditional ML algorithms. Compared to these works, the proposed system integrates threat sharing mechanisms and real-time feedback loops, extending beyond detection to proactive collaboration. Moreover, Mishra et al. (2022) emphasized ML in IoT threat intelligence, aligning with the system's RESTful API integration for interoperability.

This holistic research approach depicts a systematic procedure in designing an effective intelligent cyber threat intelligence system employing artificial neural networks. Using the CICIDS dataset and adhering to data preprocessing technique, model training and evaluation the proposed model performs successful detection and classification of the cyber threats. The approach to learning is continuous, alongside threat intelligence sharing and the clear flexibility offered by strong visualization makes the fortification of the system fluid and capable of adaptation to new threats. The incorporation of user experiences and collaborative structures improves the applicability of the solution and supports it as a benefit to organizations in the protection of the networks against cyber threats.

5. Conclusion

Limitations and Future Work: Although the proposed system showed promising results using the CICIDS dataset, it is limited by its reliance on a single dataset for training and validation. The model has not yet been tested in real-time environments or across distributed networks. Future work will focus on integrating federated learning techniques, real-time anomaly detection, and edge-computing deployment to enhance responsiveness and generalizability.

The advancement in the practice of cybersecurity involves the creation of an intelligent mechanism for cyber threat intelligence sharing and cooperation using artificial neural networks, or ANNs. Through the use of CICIDS dataset, this research has demonstrated the ability of ANNs in the earliest detection and categorization of different cybersecurity threats such as DDoS and infiltration. The proposed model improved the accuracy level and other performance indicators and this was due to the use of adequate approach of data preprocessing and construction of layers and model evaluation

methods. This performance demonstrates that ANNs are capable of identifying intricate patterns in network traffic, which would greatly improve the capacity of organisations to prevent cybersecurity threats.

Also, an incorporation of threat intelligence sharing mechanism and collaboration framework supports the group approach to cybersecurity, allowing organizations to get much more from the common threat intel. The less important data the researchers store, the more secure the system's system will be an added bonus is that it makes learning continuous, assisting the model in responding to new threats that have not been seen before over time. The additional elements of easy-to-use visualization and reporting also greatly enable the security teams to make the right decisions quickly. In conclusion, this work examines the importance of utilizing intelligent systems in present-day cybersecurity and calls for further cooperation to promote efficient and comprehensive use of such technologies as a way to improve the situation regarding cyber defence.

References

[1]. Lee, Jonghoon, et al. "Cyber threat detection based on artificial neural networks using event profiles." *Ieee Access* 7 (2019): 165607-165626.

[2]. Ahmed AA, Hasan MK, Aman AH, Safie N, Islam S, Ahmed FR, Ahmed TE, Pandey B, Rzayeva L. Review on hybrid deep learning models for enhancing encryption techniques against side channel attacks. IEEE Access. 2024 Jul 19.

[3]. Reddy, Premkumar, Yemi Adetuwo, and Anil Kumar Jakkani. "Implementation of Machine Learning Techniques for Cloud Security in Detection of DDOS Attacks." *International Journal of Computer Engineering and Technology(IJCET)* 15.2 (2024).

[4]. Ahmed, Amjed Abbas, et al. "Secure AI for 6G Mobile Devices: Deep Learning Optimization Against Side-Channel Attacks." *IEEE Transactions on Consumer Electronics* (2024).

[5]. Jakkani, Anil Kumar, Premkumar Reddy, and Jayesh Jhurani. "Design of a Novel Deep Learning Methodology for IOT Botnet based Attack Detection." *International Journal on Recent and Innovation Trends in Computing and Communication Design* 11 (2023): 4922-4927.

[6]. Mohammed AL-Ghuribi, S., Salman Ibraheem, A., Abbas Ahmed, A., Kamrul Hasan, M., Islam, S., Hafizah Mohd Aman, A., & Safie, N. (2024). Navigating the Ethical Landscape of Artificial Intelligence: A Comprehensive Review. *International Journal of Computing and Digital Systems*, *16*(1), 1-11.

[7]. Jakkani, Anil. (2024). Real-Time Network Traffic Analysis and Anomaly Detection to Enhance Network Security and Performance: Machine Learning Approaches. Journal of Electronics Computer Networking and Applied Mathematics. 4. 2799-1156. 10.55529/jecnam.44.32.44.

[8]. Muhammed, A. A., Mutasharand, H. J., & Ahmed, A. A. (2023, December). Design of Deep Learning Methodology for AES Algorithm Based on Cross Subkey Side Channel Attacks. In *International Conference on Cyber Intelligence and Information Retrieval* (pp. 355-366). Singapore: Springer Nature Singapore.

[9]. Trifonov, Roumen, Ognyan Nakov, and Valeri Mladenov. "Artificial intelligence in cyber threats intelligence." *2018 international conference on intelligent and innovative computing applications (ICONIC)*. IEEE, 2018.

[10]. Ahmed, Amjed Abbas, et al. "Efficient Convolutional Neural Network Based Side Channel Attacks Based on AES Cryptography." *2023 IEEE 21st Student Conference on Research and Development (SCOReD)*. IEEE, 2023.

[11]. Salem, Maher, and Abdel-Karim Al-Tamimi. "A novel threat intelligence detection model using neural networks." *IEEE Access* 10 (2022): 131229-131245.
[12]. Ahmed, Amjed Abbas, et al. "Design of Lightweight Cryptography based Deep Learning Model for Side Channel Attacks." *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, 2023.
[13]. Komar, Myroslav, et al. "Intelligent cyber defense system using artificial neural network and immune system techniques." *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*. Cham: Springer International Publishing, 2016.
[14]. Ahmed, Amjed Abbas, et al. "Optimization Technique for Deep Learning Methodology on Power Side Channel Attacks." *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, 2023.
[15]. Trifonov, Roumen, et al. "Artificial intelligence methods for cyber threats intelligence." *International Journal of Computers* 2 (2017).
[16]. Ahmed, Amjed Abbas, et al. "Detection of Crucial Power Side Channel Data Leakage in Neural Networks." *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, 2023.
[17]. Montasari, Reza, et al. "Application of artificial intelligence and machine learning in producing actionable cyber threat intelligence." *Digital forensic investigation of internet of things (IoT) devices* (2021): 47-64.
[18]. Vegesna, Vinod Varma, and Ashwin Adepu. "Leveraging Artificial Intelligence for Predictive Cyber Threat Intelligence." *International Journal of Creative Research In Computer Technology and Design* 6.6 (2024): 1-19.
[19]. Mishra, Shailendra, Aiman Albarakati, and Sunil Kumar Sharma. "Cyber threat intelligence for IoT using machine learning." *Processes* 10.12 (2022): 2673.
[20]. Alsaedi, Mohammed, et al. "Cyber threat intelligence-based malicious URL detection model using ensemble learning." *Sensors* 22.9 (2022): 3373.
[21]. F. Sufi, "A global cyber-threat intelligence system with artificial intelligence and convolutional neural network," *Decision Analytics Journal*, vol. 9, 2023, Art. no. 100364, doi: 10.1016/j.dajour.2023.100364.
[22]. M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Cyber Threat Intelligence Sharing Scheme Based on Federated Learning for Network Intrusion Detection," *Journal of Network and Systems Management*, vol. 31, pp. 1–23, 2021, doi: 10.1007/s10922-022-09691-3.
[23]. D. Preuveneers and W. Joosen, "Sharing Machine Learning Models as Indicators of Compromise for Cyber Threat Intelligence," *Journal of Cybersecurity and Privacy*, vol. 1, pp. 140–163, 2021, doi: 10.3390/jcp1010008.