Detection of Network Attacks on Application Servers Using Deep Learning in IoT Environments

Niranjan W. Meegammana, Harinda Fernando
Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
niranjan.meegammana@gmail.com
harinda.f@sliit.lk

***Abstract*—Internet of Things (IoT) comprises interconnected smart devices that collect data, control systems, are increasingly used in critical infrastructure, and raise security concerns due to their inherent vulnerabilities, where conventional security measures struggle to defend IoT networks. This research explores the use of Deep Learning for detecting network attacks on IoT application servers, addressing vital security concerns. The study employing the Artificial Neural Network (ANN) DL model, achieved an impressive 93% accuracy and precision Score of 0.99, highlighting its robustness in identifying various network attack types on IoT application servers. Furthermore, it demonstrated strong performance in terms of precision, recall, and F1 scores, around 0.95, 0.92, and 0.93, respectively, showcasing the ANN model's ability to make precise predictions while minimizing false positives. These findings indicate that DL, especially the ANN model, significantly strengthens IoT security by safeguarding application servers from diverse security threats. Future work will focus on countering emerging zero-day attacks and further integrating the ANN model with application firewalls to enhance IoT security.**

Keywords - IoT, Application Server, Network Attacks, Deep Learning, Artificial Neural Network

I. INTRODUCTION

IoT is used in various domains from smart homes to Critical Infrastructure (CI) to gain efficiency and cost-effectiveness. However, inherent security weaknesses in IoT have led to increased vulnerabilities, as evident from the Mirai botnet attack in 2016 [1]. The advent of 5G networks further amplifies IoT security concerns. Protecting IoT networks from cyber attacks is challenging due to limited processing power, wide distribution, and a lack of universal security standards. Conventional security measures struggle to defend IoT networks as the architecture used when developing networked IOT systems is significantly different from traditional network architecture. The IoT security issues result in data breaches leading to privacy violations and financial losses. Disruptions to Critical Infrastructure (CI) may cause significant economic losses and endanger human lives. Businesses may incur financial losses from attacks on IoT-based services. Compromised IoT devices in healthcare and automotive can create life-threatening situations. Environmental monitoring IoT devices could be used to cause devastating consequences [2]. Implementing strong security mechanisms on IoT networks can help reduce these risks, and give greater confidence among consumers and industries to use IoT technologies, fostering growth and innovation in the IoT Industry [1].

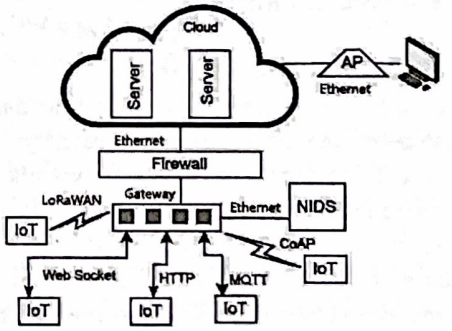

Fig. 1 An IoT environment.

A. Deep Learning

Deep Learning (DL) is a branch of Artificial Intelligence that can analyze historical traffic data to identify attack patterns and perform automated threat identification, intrusion detection, and behavior modeling. DL techniques are capable of addressing different phases of cyber attacks. During the reconnaissance, they can detect probing and malware downloads. In the weaponization phase, they can identify vulnerabilities in application servers, and during exploitation, they can detect various types of attacks [3]. Sagu et al. [4] effectively utilized the Artificial Neural Network (ANN) DL algorithm to detect IoT network attacks. In the face of evolving cyber threats to IoT networks, the integration of DL techniques into IoT application server protection presents a viable solution to safeguard CI better [1], [3], [4], [5]. Therefore, this research contributes to strengthening application server security in IoT environments, while providing a potential future direction for IoT network security research. The next part of this paper is organized into a literature review of deep learning approaches to IoT

979-8-3503-5813-1/23/$31.00 ©2023 IEEE

network attack detection, aimed at establishing a theoretical foundation, then methodology presents the research design, dataset, and approach, detailing the steps in the study. Subsequently, the paper presents the results and findings, analyzing performance comparisons, followed by a discussion and insights on future directions, leading to the conclusion.

II. LITERATURE REVIEW

A. IoT Security Challenges

The IoT devices are capable of operating autonomously, rapidly integrating into Supervisory Control and Data Acquisition (SCADA) systems in CI like power plants and hospitals. They expose numerous vulnerabilities increasing security risks, where traditional security measures struggle to protect IoT networks due to increased device exposure, limited processing power, and the absence of universal IoT security standards. A single compromised IoT device can lead to the exploitation of an entire network [6]. International political conflicts are escalating into cyber warfare evident from the Ukraine-Russia conflict targeting ioT in CI inflicting widespread disruptions.

B. Attacks on IoT Application Servers

An IoT application server handles data management, communication, device control, and security within IoT networks. IoT application server attack vectors span from DDoS attacks to HTTP floods, SQL injections, and cross-site scripting, where protecting the application servers is particularly challenging due to the constrained computing power, and use of insecure protocols such as HTTP, FTP, and telnet that open avenues for sophisticated attacks [5].

C. Machine Learning (ML) and Deep Learning

ML and DL-based intrusion detection use patterns learned from labeled data to predict impending attacks. Research indicates that supervised learning algorithms are effective in network attack detection [5]. Among them, DL stands out due to its ability to handle non-linear relationships and complex patterns and scalability, thus making it effective for unknown attack detection. Moreover, DL offers noteworthy advantages such as automatic feature extraction and adaptability to changing data distributions [2]. However, DL requires substantial computational resources, susceptible to overfitting and challenging to interpret.

D. Similar Work

Aversano et al. [7] highlight the limitations of decades-old datasets in detecting attacks on modern IoT environments. Wu et al. [8] introduce several DL applications to present problems in network security and attack detection and confirm the distinctive power of Artificial Neural Networks (ANN). Aleesa et al. [9] conducted a DL study using 1,840,046 instances of the UNSW-NB15 dataset, claiming to achieve 99.26% accuracy for binary classifications. Considering the high accuracy and substantial size of the dataset raises a legitimate concern about potential overfitting to the training data. Hence, it requires validating the models' performance with real-world data to ensure generalization. Husain et al. [10] compare XGBoost, Random Forest, and Neural Networks algorithms, and conclude that XGBoost demonstrates superior performance. However, not addressing data standardization, cross-validation, and hyperparameter tuning in the process, raises a valid concern about achieving optimal performance of resulting models. Choudhary and Kesswani [11], employed Deep Neural Network (DNN) to identify network attacks and emphasize the need for an appropriate dataset, and concluded that the UNSW-NB15 is more accurate in classification tasks among other datasets used. However, direct comparison of these datasets without incorporating real-world simulations could be problematic, as the three datasets differ in terms of their features, scale, and the contexts they were created.

E. Research Gap

Overall, several studies have underscored the limitations of current network attack detection tools and suggested various algorithms using distinct datasets. However, the consensus remains elusive on an optimal algorithm and a dataset for effective network attack detection [5]. A substantial portion of research has overlooked the importance of feature selection, data standardization, and hyperparameter tuning. Interestingly, there has been a scarcity of research focusing on DL for attack detection on IoT application servers. This research aims to bridge those gaps by utilizing the Artificial Neural Network (ANN) algorithm in DL for attack detection in IoT application servers in CI.

III. METHODOLOGY

The design of this research adheres to scientific and experimental techniques, employing quantitative analysis of data extracted from the UNSW-NB15 dataset.

. Research Process

The research followed the steps shown in Figure 2 [12], which involved threat modeling, data collection, analysis, data preparation, feature and model selection, parameter tuning, validation, model building, and testing stages.

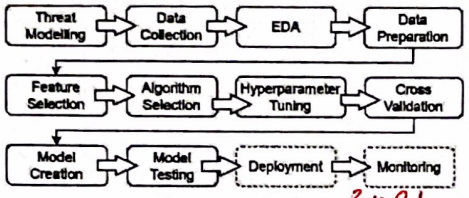

Fig. 2 Research process.

B. Threat Modelling Process

The study performed a threat modeling process using MITRE ATT&CK ICS framework to identify Tactics, Techniques, and Tools (TTP) used by adversaries, including detection and mitigation measures in IoT environments [3].

C. Data Collection Process

The UNSW-NB15 dataset selected includes 2.43 million instances of network traffic data with 42 attributes, covering 9 attack types. Instances are labeled for supervised learning tasks as benign and attack classes. Widely used in recent studies [13][14], it holds relevance for network intrusion detection in IoT environments.

D. Exploratory Data Analysis (EDA) Process

The research conducts Exploratory Data Analysis (EDA) on the dataset, visualizing distribution and correlations to gain insights into data characteristics. This aims to identify patterns and address anomalies [15].

E. Data Preparation Process

The study extracted a 10% sample from the dataset, cleaned, and balanced the target class through random undersampling [2]. The data was then split into training and testing sets at a 95:5 ratio.

F. Data Standardization Process

Data standardization is vital for ensuring consistent scaling among numerical features. This process scales and transforms the numerical values, centering them around zero. Standardization guarantees that each feature has a similar scale during model training. Data standardization used $x_{std}=(x - \mu)/ \delta$ formula, where x_{std} is the standardized value of the feature, x is the original value of the feature, μ is the mean of the feature across the dataset, δ is the standard deviation of the feature across the dataset [14].

G. Feature Selection Process

This process identified significant features in the dataset, using wrapper and filter methods [14] as shown in Table I. The results were amalgamated to determine the optimal features aligning with the research objectives.

TABLE I
FEATURE SELECTION APPROACH

Feature Selection			Algorithms			
	Method		LOGR	SVM	KNN	ANN
Wraper	SelectKBest	☐	☑	☑	☑	☑
	Random Forest Importance	☐	☑	☑	☑	☑
	Pearson Corelation	☐	☑	☑	☑	☑
	Forward Feature Selection	☐	☑	☑	☐	☐
Filter	Information Gain	☑	☐	☐	☐	☐
	Varince Threshold	☑	☐	☐	☐	☐
	Lasso Regression	☑	☐	☐	☐	☐

H. Algorithm Selection Process

The research systematically prototyped multiple ML and DL learning algorithms to identify the most suitable model for research objectives. Evaluated algorithms included Logistic Regression (LR), Decision Trees (DT), Random Forests (RF), XGBoost, and K-Nearest Neighbors (KNN) in ML, and Artificial Neural Networks (ANN) in DL.

I. Performance Measurement of Algorithms

Accuracy, Precision, Recall, F1 Score, and Mean Squared Error, along with the Receiver Operating Characteristic (ROC) curve, are key metrics in ML and DL performance assessment. Accuracy defines the ratio of correctly predicted instances to the total dataset. Precision gauges the model's ability to avoid false positives, while Recall represents the ratio of true positive predictions to actual positives. The F1 score is the harmonic mean of precision and recall [15]. The ROC curve illustrates a binary classifier's diagnostic ability, showing the trade-off between sensitivity and specificity with varying thresholds. Formulas for these metrics are used to compute performance [16].

$$Accuracy = Number\ of\ Correct\ Predictions/Total\ Number\ of\ Instances.$$

$$Precision = True\ Positives\ /(True\ Positives + False\ Positives)$$

$$Recall = True\ Positives/(True\ Positives + False\ Negatives)$$

$$F1\ Score = 2 * Precision * Recall/(Precision + Recall)$$

Fig. 3 Performance evaluation formulas.

J. Hyperparameters Tuning Process

The hyperparameters help find the most appropriate settings for the model to achieve optimal model performance. The research employed Random Search and Grid Search techniques to obtain the best hyperparameters [17]. The resulting hyperparameters were used to build two ANN models. The results section contains the hyperparameters and performance information.

K. K-Fold Cross Validation Process

This process validated the model's generalization on an unseen dataset, adding evidence of credibility to the research through a rigorous evaluation of performance.

L. Model Building and Training Process

Two ANN models were built using two optimized hyperparameter sets using the ANN algorithm. The training utilized a set containing 95% of the data, enabling the model to learn from labeled data to capture patterns distinguishing benign traffic from attacks.

M. ANN Model Testing Process

The test dataset, comprising 5% of instances, was employed to assess the generalization ability of the trained ANN model. Notably, the model had not encountered the testing dataset during training. The results of the model testing performance are presented in the results section.

IV. RESULTS and DISCUSSION

A. Dataset Preparation Results

Applying undersampling to the extracted 10% sample dataset, addressed the class imbalance between benign and attack instances to enable equal learning by the model. It reduced the dataset to 186,000 samples, split at a 95:5 ratio as shown in Table II for training and testing. Given the dataset's size, the 95:5 split allocated more data for training, potentially enhancing the model's generalization.

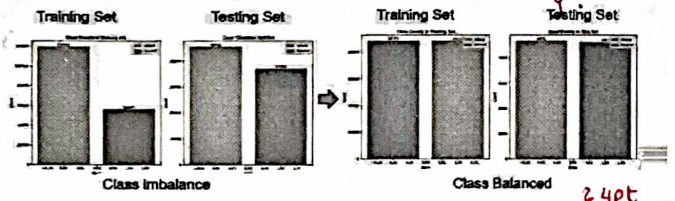

Fig. 4 Balancing of imbalanced classes in the dataset.

TABLE II
DATA DISTRIBUTION AFTER DATA PREPARATION

Dataset	Split	Samples	Training		Testing	
			Attack	Benign	Attack	Benign
Imbalanced	68:32	257,673	68%	32%	55%	45%
Balanced	95:05	186,000	50%	50%	50%	50%

B. Feature Selection Results

This process diminished the original 42 features to 20 using various wrapper and filter techniques, along with results amalgamation. This reduction aims to improve model performance and alleviate model complexity [14].

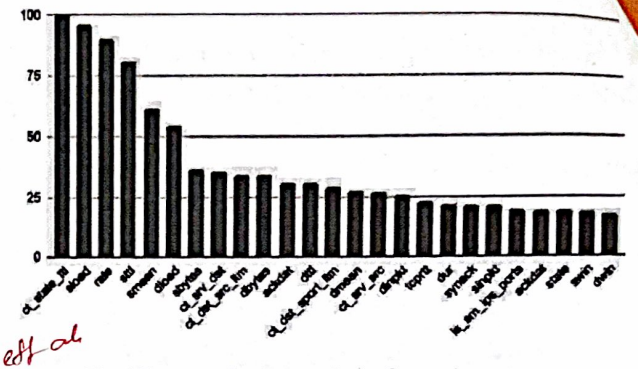

Fig. 5 Feature selection results by feature importance.

C. Algorithms Evaluation Results

The study assessed a range of ML algorithms, encompassing Logistic Regression, Naive Bayes, Decision Trees, Random Forest, XGBoost, Ensemble, K-nearest Neighbour (KNN), and ANN. The dataset was used to obtain performance metrics, training, and testing times for network attack detection. The results are presented below.

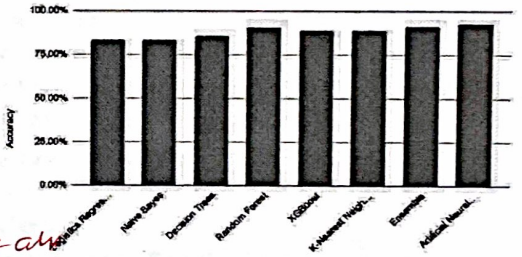

Fig. 6 Classifiers and accuracy results.

TABLE III
CLASSIFIER PERFORMANCE

Classifier	Accuracy	Precision	Recall	F1 Score	Training Time	Testing Time
Logistics Regression	83.58%	79.85%	89.60%	84.45%	4.2231	0
Naive Bayes	83.58%	82.05%	80.14%	81.08%	4.2231	0
Decision Trees	85.53%	84.79%	86.41%	85.59%	2.2434	0
Random Forest	90.89%	87.97%	94.64%	91.18%	23.8606	0.1251
XGBoost	88.90%	88.40%	89.43%	88.91%	7.9519	0
K-Nearest Neighbors	88.90%	88.40%	89.43%	88.91%	0.5511	0.0169
Ensemble	91.22%	93.77%	88.20%	90.90%	32.4454	26.1635
Artificial Neural Network	92.88%	93.81%	91.74%	92.77%	79.6898	0.5802

Classifier performance evaluation results are shown in Table III. The ANN algorithm outperformed others in detecting network attacks with accuracy, precision, recall, and F1 score. Although its training time was slightly higher, the difference was negligible in the context of the study.

D. Hyperparameters Tuning Results

The hyperparameter tuning process, employing Random Search and Grid Search techniques, yielded the hyperparameters for training two ANN models shown in Table IV.

TABLE IV
HYPERPARAMETER TUNING RESULTS

Random Search	Value	GridSearch CV	Value
neurons_layer1	256	neurons_layer1	256
activation_layer1	relu	activation_layer1	relu
neurons_layer2	128	neurons_layer2	256
activation_layer2	relu	activation_layer2	relu
learning_rate	0.001	learning_rate	0.001
dropout_hidden	0.0	dropout_hidden	0.2

Table IV offers insights into the performance of hyperparameter tuning, indicating that the Random Search technique slightly outperformed the Grid Search.

TABLE V
HYPERPARAMETER TUNING PERFORMANCE

Random Search	Value	GridSearch CV	Value
Validation Accuracy	94.0%	Validation Accuracy	90.0%
Precision	94.0%	Precision	89.0%
Recall	92.0%	Recall	90.0%
F1 Score	93.0%	F1 Score	90.0%
Mean Squared Error (MSE)	0.07	Mean Squared Error (MSE)	0.07
R-Squared (R2):	0.84	R-Squared (R2):	0.73
Mean Absolute Error (MAE)	: 0.06	Mean Absolute Error (MAE)	: 0.07
ROC-AUC Score	0.99	ROC-AUC Score	0.90

In hyperparameter tuning the ROC for Random Search and Grid Search results yielded 0.99 and 0.90, respectively. Notably, Random Search achieved a superior ROC.

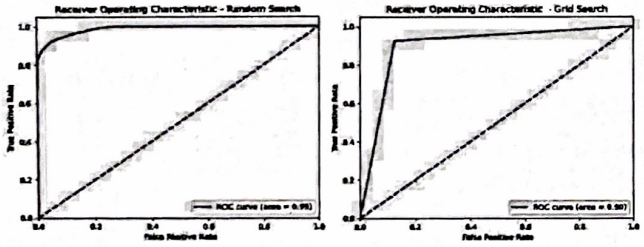

Fig. 7 Hyperparameter tuning results.

E. K-Fold Cross Validation Results

The model validation results, achieving a mean ROC score of 0.99, provide evidence of data variability, model reliability, and generalization capability, ensuring low bias and variance. Ultimately, this assures credibility to the research.

TABLE VI
K-FOLD CROSS VALIDATION RESULTS

Metric	Value
Mean ROC AUC	0.99
Mean Accuracy	0.93
Mean Precision	0.94
Mean Recall	0.92
Mean F1 Score	0.93

E. Model Training and Testing Results

The Model Training and Testing using the hyperparameters from both Random Search and Grid Search were utilized to train two ANN models on a training dataset. Subsequently, both ANN models were tested using a separate testing dataset, yielding the results shown in Table VII.

TABLE VII
ANN MODEL TESTING RESULTS

Random Search	Value	GridSearch CV	Value
Validation Accuracy	0.93	Validation Accuracy	0.93
Precision	0.94	Precision	0.96
Recall	0.92	Recall	0.89
F1 Score	0.93	F1 Score	0.92
ROC-AUC Score	0.98	ROC-AUC Score	0.99
Average Precision Score	0.99	Average Precision Score	0.99

The ROC for the two ANN models, crafted with hyperparameters from Random Search and Grid Search, recorded 0.98 and 0.99, respectively, and a precision Score 0.99.

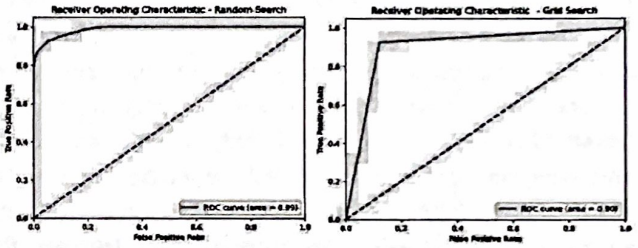

Fig. 8 ROC curves for ANN models.

These results offer comprehensive insights into the ANN models created by showcasing their overall performance and capability to handle IoT application server attack classification tasks. The findings signify that the ANN model can achieve a high proportion of accurate predictions effectively, leveraging learned patterns and relationships within the data. As a result, it demonstrates the capacity to generalize well to new, unseen instances.

F. Discussion

The results indicate the effectiveness of the Artificial Neural Network (ANN) DL model in identifying network attacks on IoT application servers. The higher accuracy, precision, recall, and F1 score achieved demonstrates the capacity of Deep Learning to identify complex patterns and the capability to adapt through rigorous training. Other classification algorithms showed network attack detection possibilities. However, ANN performed better across all metrics with higher accuracy, robustness, and consistency. The variable training and testing arrangements highlighted the trade-offs between model performance and efficiency for scalability. Accuracy and response speed are important in

network attack detection, where algorithm selection is paramount. Despite certain benefits, the research has likely imperfections, such as the use of synthetic data and untested measures. Future research will focus on evaluating new vectors of attack and hyperparameter tweaking. The successful application of the ANN model in this study in attack detection promises enhanced application server security in IoT environments.

V. CONCLUSION

In conclusion, the research demonstrates that the Artificial Neural Network (ANN) Deep Learning model can be used to detect IoT application server attacks effectively. The achievement of an accuracy of 93%, showcases the robustness of the ANN model. Furthermore, precision, recall, and F1 scores performance data averaging around 0.95, 0.92, and 0.93 respectively, and the exceptional Precision Score of 0.99 on accurate true positive predictions, emphasize the ANN model's ability to balance fewer false positives with more precise identifications and classifying various attack types to make precise predictions for protecting application servers in IoT environments against a range of security threats. The future work of this study involves countering emerging zero-day attacks on IoT application servers by employing the ANN DL model created and integrating with an application firewall. This approach will leverage the capabilities of the DL model to enhance the security of IoT environments and adapt to new attack scenarios.

~~VI.~~ ACKNOWLEDGEMENT

I would like to express my sincere thanks to Dr. Harinda Fernando, my research supervisor, and Dr. Anuradha Jayakody, the head of graduate studies, for the guidance, insights, and support throughout this research.

~~VII.~~ REFERENCES

[1] K. Rose, S. Eldridge, and L. Chapin, "The Internet of Things: An Overview Understanding the Issues and Challenges of a More Connected World," Oct. 2022.

[2] N. Moustafa, M. Abdel-Basset, and R. Mohamed, *Deep Learning Approaches for Security Threats in IoT Environments*. Wiley-IEEE Press, 2022.

[3] M. Bagaa, T. Taleb, J. B. Bernabe, and A. Skarmeta, "A Machine Learning Security Framework for Iot Systems," *IEEE Access*, vol. 8, pp. 114066–114077, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2996214.

[4] A. Sagu, N. S. Gill, and P. Gulia, "Artificial Neural Network for the Internet of Things Security," *International Journal of Engineering Trends and Technology*, vol. 68, no. 11, pp. 129–136, Nov. 2020, doi: https://doi.org/10.14445/22315381/ijett-v68i11p218.

[5] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: a comprehensive survey," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, p. 154851292095127, Sep. 2020, doi: https://doi.org/10.1177/1548512920951275. [6] A. Almalawi, Z. Tari, A. Fahad, and X. Yi, *SCADA security: machine learning concepts for intrusion detection and prevention*. Hoboken, NJ, USA: Wiley, 2021.

[7] L. Aversano, M. L. Bernardi, M. Cimitile, and R. Pecori, "A Systematic Review on Deep Learning Approaches for IoT Security," *Computer Science Review*, vol. 40, p. 100389, May 2021, doi: https://doi.org/10.1016/j.cosrev.2021.100389

[8] Y. Wu, D. Wei, and J. Feng, "Network Attacks Detection Methods Based on Deep Learning Techniques: a Survey," *Security & Communication Networks*, pp. 1–17, Aug. 2020, doi: https://doi.org/10.1155/2020/8872923.

[9] A. Aleesa, M. Y. Thanoun, A. A. Mohammed, and S. Nan, "Deep-Intrusion Detection System with Enhanced UNSW-NB15 Dataset Based on Deep Learning Techniques," in *Journal of Engineering Science and Technology*, University of Mosul Ninevah University Nan M Sahar, Feb. 2021, pp. 711–727.

[10] A. Husain, A. Salem, C. Jim, and G. Dimitoglou, "Development of an Efficient Network Intrusion Detection Model Using Extreme Gradient Boosting (XGBoost) on the UNSW-NB15 Dataset," *IEEE Xplore*, Dec. 01, 2019. doi: https://doi.org/10.1109/ISSPIT47144.2019.9001867.

[11] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD, and UNSW-NB15 Datasets using Deep Learning in IoT," *Procedia Computer Science*, vol. 167, pp. 1561–1573, 2020, doi: https://doi.org/10.1016/j.procs.2020.03.367

[12] S. Raschka and Vahid Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, sci-kit-learn, and TensorFlow*. Birmingham (UK): Packt Publishing, 2017.

[13] N. Moustafa, "The UNSW-NB15 Dataset | UNSW Research," *research.unsw.edu.au*, 2021.

[14] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *Journal of Big Data*, vol. 7, no. 1, Nov. 2020, doi: https://doi.org/10.1186/s40537-020-00379-6.

[15] C. Wang and D. Szeto, *Engineering Deep Learning Systems*. Manning, 2022.

[16] E. Tsukerman, *Machine learning for cybersecurity cookbook: over 80 recipes on how to implement machine learning algorithms for building security systems using Python*. Birmingham; Mumbai: Packt Publishing, 2019.

[17] T. Janssen, "Hyperparameter tuning for Artificial Neural Networks applied to inverse mapping parameter updating," Jun. 2022.

~~It~~ ~~Plz~~ Double check the format of References. Make sure they adhere to IEEE format