MALWARE DETECTION OF IOT NETWORKS USING MACHINE LEARNING: AN EXPERIMENTALSTUDY WITH EDGE IIOT DATASET

Nizwan Hamza
School of Computing and Engineering
University of Gloucestershire
Cheltenham, United Kingdom
e-mail: s4213890@glos.ac.uk

MWP Maduranga
Department of Computer Engineering
General Sir John Kotelawala Defence University
Ratmalana, Sri Lanka
e-mail: pasanwellalage@kdu.ac.lk

H.K.I.S Lakmal
Department of Mechatronic and Industry Engineering
NSBM Green University Town
Homagama, Sri Lanka
e-mail: isuru.l@nsbm.ac.lk

R.P.S Kathriarachchi
Department of Information Technology
General Sir John Kotelawala Defence University
Ratmalana, Sri Lanka
e-mail: pathum@kdu.ac.lk

***Abstract*— Artificial Intelligence (AI) is a powerful tool that assists security operation analysts in addressing threats in sectors like IIoT and IoT. As cyber-attacks increase, proactive security measures are crucial to protect critical infrastructure and sensitive data. AI-based learning models provide deeper insights into cyber threats, aiding in the development of effective mitigation strategies. This work contributes towards using supervised Machine Learning (ML) algorithms for Internet of things(IoT) sensor data to identify vulnerabilities. Supervised ML algorithms, including KNN, DTC, LR, SVM, and RFC, are evaluated using the "Edge-IIoTset" dataset for malware detection. Results show that RFC achieves 94% accuracy, outperforming other algorithms.**

I. INTRODUCTION

Implementing malware detection systems in IIoT systems is important to prevent and detect malicious software that can compromise the integrity, availability, and confidentiality of the system, safeguarding industrial operations and sensitive data from potential cyber threats. The use of contemporary digital technologies in security professionals' daily work is expanding quickly. Artificial Intelligence (AI) is being used more and more by businessesin the security sector to handle the complexity of their security processes, which must adapt to the current data flood and stay up to client expectations. In the field of computer science, AI is currently buried in several facets of security [1]. AI is used in many aspects of security to keep ussecure, including home security, video surveillance, and threat assessment for major events. In other words, AI serves as the fundamental driving force enabling security systems to exhibit intelligent behavior independently, ensuring safety even in the absence of human oversight. Security systems need to have human-like intelligence to carry out jobs and address security issues. Due to the capabilities of AT, it is possible to mimic human intelligence in security systems. AI developers simulate human intelligence in security technologies using mathematical operations and intricate algorithms. Many people are unaware that psychology, neurology, philosophy, and communications are also major influences on AI. Computer science can be viewed as merely a facilitator, a way to implement AI.

There are two distinct types of AI security, so-called strong AI (sometimes known as "superintelligence") and weak AI, depending on their respective objectives [2]. Weak AI security solutions function at a relatively low degree of intelligence, while strong AI seeks to match or even surpass the intellectual prowess of humans by mimicking human reasoning and behavior. Although they do not gain a thorough comprehension of the issues to be solved, the security firms that use them are already benefiting much fromtheir ability to carry out duties autonomously and automatically. Alert management, which increases alert verification accuracy and lowers the number of false alarms,is a well-known example of strong AI in security.

Even though "machine learning," and "deep learning" are subsets of "artificial intelligence," sometimes those terms are used interchangeably. Artificial Intelligence (AI) is merely the general term and more expansive definition that covers both machine learning and deep learning [3]. Security system developers, however, can choose the proper AI-based tool for a certain security measure and make better decisions when searching for security solutions by being aware of which features differ and their unique capabilities.

A. Machine Learning

A subset of artificial intelligence called machine learning (ML) which is rapidly gaining popularity in the security sector, is defined as "the capacity for computers to learn without being explicitly programmed" [4]. It speaks about systems that learn on their own, without prior programming,

based on experience. A lot of security systems, including SIEM and malware analysis, use clever self-learning algorithms as the foundation for their engines. By identifying patterns and specific regularities in massive amounts of data and intelligently linking and correlating them, ML applications can conclude and make predictions.

They look to differentiate between typical and abnormal behavior to independently identify dangers and comprehend security situations. The software that powers this AI looks through enormous volumes of data for specific patterns and regularities before automatically classifying and correlating security occurrences. Systems having machine learning capabilities will get smarter over time and perform better if exposed to more data streams. This is how ML security applications can help free up resources and relieve staff members of tiresome, pointless, and repetitive activities [5].

B. Deep Learning

Systems adopting DL can repeatedly link what they have learned with fresh stuff based on the information at hand and so continuously learn new things. These systems can then make independent decisions, foresee the future, and at some point, critically evaluate those decisions. The main distinction between deep learning applications and machine learning is that the latter often learn without human intervention. The systems based on DL mechanisms are capable of making exact judgments without human intervention using programmable neural networks. When forecasting customer behavior using speech, object, photo, or face recognition capabilities, deep learning is extremely helpful. Deploying DL in situations where substantial amounts of data need to be examined for underlying patterns is done for this reason [6].

II. AI IN CYBER SECURITY

AI has several distinctive qualities that enable it to advance in the field of security. Physical and cyber security are just two of the many applications where artificial intelligence is expanding quickly [7].

The convergence of various variables is what makes this emergence possible and what propels it: In the first place, the accessibility of a range of data [8] (referred to as “big data”) enables security experts to train their algorithms efficiently. For instance, many sensors used in physical security, such as thermal, infrared, and motion detectors in video image surveillance cameras, generate this data. Open-source software, the rise of deep neural networks, and next-generation computing architecture are additional factors accelerating the development of AI in security.

Enterprises are more susceptible to cyber security risks due to their connections to several independent endpoints. The expansion of constantly connected businesses, gadgets, and applications only serves to further support this. AI and software empower their users to safeguard any system by giving security professionals proactive threat mitigation tools like real-time notifications. AI’s machine learning capabilities enable proactive malware and threat prevention rather than merely detection, greatly enhancing AI’s potential and value proposition in the security industry.

AI offers the vital threat analysis and identification capabilities that cybersecurity professionals must adopt to lower the risk of breaches and improve security posture because humans are no longer capable of fully defending an organization’s dynamic threat surface.

AI enables security personnel to protect lives and property more effectively while using fewer human resources in control rooms: By incorporating AI into surveillance systems, humans may avoid doing tedious or repetitive tasks like sorting through hours of video material while simultaneously increasing the efficiency and accuracy ofsurveillance by using threat recognition at machine speed.

It is frequently difficult for modern security analysts to find the time to identify new dangers. They also frequently employ time- and resource-consuming danger-hunting techniques, which frequently leads to alert fatigue[9]. Traditional security methods expect and require more human involvement than AI security aims to, which increases efficiency by reducing the amount of time spent looking for threats.

AI can assist security experts in identifying and prioritizing hazards in the context of security. It can assist them in managing incident response, quickly identifying malware on a network, and stopping incursions in their tracks. AI is used in physical security to improve the responsiveness, potency, and efficiency of remote monitoring, perimeter security systems, and access control. How? AI significantly aids remote monitoring agents by spotting unexpected circumstances, such as specific movement in a location where there shouldn’t be any movement at that moment[10]. These agents can react more swiftly because they can more immediately comprehend the situation on the ground. Additionally, AI can differentiate between general and human movements, which minimizes false alerts. Moreover, to be a rapidly expanding trend across many industries, AI can intelligently link data, make independent and automatic decisions, and automate forecasts. This opens up new possibilities. AI appliances shouldn’t be overlooked if you want to be prepared for the future, especially in industries like security where massive amounts of data must be handled in relevant ways. The cybersecurity sector will continue to be impacted by AI in the future. It will be fascinating to watch how this impact plays out over the coming years. Deep-learning technologies are expected to provide unmatched insight into human behavior, enabling video systems to monitor and forecast crime more effectively, according to AI device manufacturers and security researchers. AI will probably become more adept atoffering scalable solutions in different vertical markets [11].

III. DATASET

In this research, we propose the utilization of Edge-IIoT dataset, a cyber security dataset of IoT and IIoT applications, by machine learning-based intrusion detection systems [15]. The "Edge-IIoTset Cyber Security Dataset of IoT & IIoT" is an extensive dataset designed specifically to assess and improve cybersecurity techniques in the realm of Internet of Things (IoT) and Industrial Internet of Things (IIoT) environments. Its primary objective is to provide researchers

and practitioners with a diverse and realistic dataset that can be used to analyze, detect, and mitigate cyber threats targeting IoT and IIoT systems.

In this dataset, there are 63 variables with 157,800 observations. Focusing on the Analysis, the qualitative output(response) variable is "Attack_Type" and the rest are quantitative input variables. There are no missing values in the data set. In this dataset Man in the middle (MITM), Fingerprinting, Ransomware, Uploading, SQL_injection, DDoS_HTTP, DDoS_TCP, Password, Port_Scanning, Vulnerability_scanner, Backdoor, Cross Site Scripting (XSS), DDoS_UDP, DDoS_ICMP are the identified threats. All 14 Attack Types have been divided into 5 categories as DDOS, Injection, Malware, MITM, and Information Gathering.

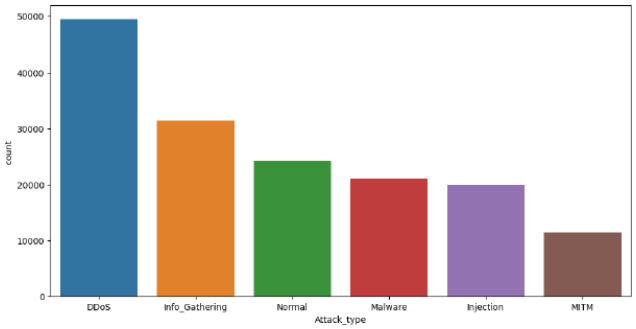

Figure 1: Distribution of types of attacks

Figure 1. shows each count of attack types. The attack types are: "DDOS," "Info_gathering," "Normal," "Malware," "Injection," and "MITM." The y-axis represents the count of each attack type. The the "DDOS" attack type is the tallest, indicating that it has the highest sample count, nearly 50,000. The "Info_gathering" attack type has the next highest count, nearly 30,000, followed by the "Normal" attack type with a count of nearly 25,000. The "Malware" and "Injection" attack types have counts greater than 20,000, while the "MITM" attack type has a count of nearly 12,000.

IV. MACHINE LEARNING ALGORITHMS

Programming a machine so that it learns from its experiences and many instances without being explicitly taught are known as machine learning. It's an application of AI that enables autonomous machine learning. Machine learning algorithms combine logic and mathematics to adapt to changing input data and perform better over time. The majority of algorithms are written in Python because of their versatility in handling various machine-learning jobs [16].

To demonstrate the given dataset and identify IoT and IIoT network intrusions, we have utilized the machine learning algorithms listed below in this research

A. k- Nearest Neighbors (KNN)

A straightforward but effective machine learning approach which is utilized for both classification and regression problems. In KNN, the training dataset's k nearest neighbors are taken into account when making a prediction for a new data point. The technique uses a distance metric to calculate the distance between each training data point and the new data point. The forecast is then made via a majority vote for classification tasks or by averaging the target values for regression tasks using the k nearest neighbors. KNN is a non-parametric algorithm, which means it makes no assumptions about the distribution of the underlying data.

KNN is easily transferable to security systems and detecting security threats in IoT and IIoT systems [17].

B. Decision Tree Classification Algorithm (DTC)

It builds a model that resembles a tree, with each internal node standing in for a feature or attribute, each branch for a decision rule, and each leaf node for a class label. Using measures like Gini impurity or information gain, the DTC algorithm iteratively divides the training data based on the feature that offers the best split. Until a stopping requirement is satisfied, such as reaching a maximum depth or a minimum number of samples per leaf, this process is repeated. When making a prediction, the algorithm iterates through the decision tree using the feature values of the input data, assigning a class label based on the leaf node it reachesin the process. Both numerical and categorical features can be handled by decision trees, and they are comprehensible [18].

C. Logistic Regression (LR)

This supervised machine learning approach is popular for predicting binary or multi-class discrete values. The probability of an event occurring is predicted using this approach by fitting the data into a logistic curve or logistic function. Because of this, it is also known as logistic regression.

The Sigmoid function, commonly known as logistic regression, converts any real-valued number to a value between 0 and 1. This algorithm is commonly used to identify spam emails, forecast website or ad clicks, and trackclient attrition [20].

D. Support Vector Machines (SVM)

A Support Vector Machine (SVM) is a popular and powerful machine learning algorithm used for classification tasks. The SVM algorithm aims to find an optimalhyperplane that separates data points belonging to different classes with the maximum margin. It works by transformingthe input data into a high-dimensional feature space and thenfinding the hyperplane that best separates the classes. SVMs are particularly effective in handling complex data and can handle both linearly separable and non-linearly separable datasets using kernel functions. The decision boundary generated by an SVM is based on support vectors, which are the data points closest to the separating hyperplane. SVMs are known for their ability to generalize well, handle high-dimensional data, and robustly handle outliers [22].

E. Random Forest Classifire (RFC)

Random forest is a type of ensemble learning algorithm that can be applied to both classification and regression problems. It is known for its flexibility and user-friendly nature. The algorithm works by constructing decision trees based on the provided data samples. It then generates predictions from each tree and determines the final solution through a voting process. Additionally, the random forest algorithm provides valuable insights into the importance of different features in the data. By combining multiple decision trees, the random forest algorithm forms a collection of trees known as a forest, which gives the algorithm its name. In the case of the random forest classifier, increasing the number of trees in the forest typically leads to higher accuracy in the predictionsprovided adequate number of training samples are available.

V. MODEL TRAINING

The model training process was implemented using Python, leveraging the Jupyter Notebook IDE. Jupyter Notebook is a widely used tool for developing Python machine learning models, offering an interactive environment for code writing and execution within a web browser. It facilitates seamless code documentation, data visualization, and collaborative work among researchers. To initiate the development of machine learning models within Jupyter Notebook, the first step involves installing essential Python libraries for machine learning and data analysis. In this study, the Numpy, Pandas, and Matplotlib libraries were employed to support various data manipulation and visualization tasks within the model.

A. Dataset

The dataset was loaded to a Comma Separated Value (CSV) file. It was loaded to the python environment using the 'read_csv' function offered by the Pandas library, and the loaded dataset is in a format of a Pandas Data Frame. The CSV file contains various columns, and some columns contain mixed data types. The program may show a warning about this hence it was required to convert such columns to 'str' data type during the import process itself. The last column in the loaded Data Frame shows the attack type using various inputs taken from other columns.

Also, it was identified the several columns as unnecessary due to invalid and unrelated data. For example, when we analyze the attack type, some inputs such as time stamp and IP address are irrelevant. At the same time, if a column contains thousands of different data, encoding cannot bedone. I have used the pandas reference number in the code toomit them. The below list contains all omitted columns with the reference number with the column name.

- 0 – frame.time: The date and time stamp of the data packet cannot impact the result attack type
- 1 – ip.src_host: IP cannot impact the attack type
- 2 – ip.dst_host: IP cannot impact the attack type
- 11 – http.file_data: Too many data hence encoding is impossible.
- 13 – http.request.uri.query: Too many data hence encoding is impossible
- 16 – http.request.full_uri: Too many data hence encoding is impossible
- 31 – tcp.options: Too many data hence encoding is impossible
- 32 – tcp.payload: Too many data hence encoding is impossible.

B. Encoding

Encoding means converting it from its original categorical form into a numerical form. For example, if the dependent variable is a categorical variable with values "low", "medium", and "high", encoding it could involve converting it to numerical values such as 0, 1, and 2, respectively. In our case, many of the dependent variables and independent variables are in categorical form and need tobe encoded before any further processing.

We have created a new variable 'encodable X' with all dependent variables to be encoded and applied 'LabelEncoder' class from the 'sklearn.preprocessing' to make them numerical. The same class is applied to dependent variable also after defining 'X' and 'y'.

C. Training and Testing Set

Data is typically divided into two sets in machine learning: the training set and the testing set.

The model is trained on the training set, and its performance is assessed on the testing set. The plan is to learn the patterns and correlations in the data using thetraining set, and then test the model's generalizability to fresh, untested data using the testing set.

With a standard split of 70–30 or 80–20, the data are often divided at random into training and testing sets. For instance, 80% of the data in my code is utilized for training, while the remaining 20% is used for testing. To divide the data into training and testing sets, I used the scikit-learn library's 'train_test_split()' function. The 'test_size' argument indicates how much of the data should be used fortesting, while the 'random_state' option establishes the reproducible random seed.

D. Feature Scaling

Scaling the features in machine learning is frequently necessary to ensure that they are on the same scale and have the same range. Scaling the features can be accomplished using the 'StandardScaler' class from the'scikit-learn'library. We created a 'StandardScaler' object and fit it to thefeatures using the 'fit()' method. The 'fit()' method computes the mean and standard deviation of each feature, which are used to scale the features.

The features are then transformed using the "transform()" method, and the transformed features are then stored in new variables called "X_train" and "X_test." The scaled features now have a mean of 0 and a standard deviation of 1, and they are all on the same scale.

E. Model Training

Once the data preprocessing was completed, the model training was carried out by initializing kNN, DTC, LR, SVM and RFC models, and then fitting the data to these models. Further details of model training are available at https://github.com/isurushanaka/NIDS-EDGE_IIoT. At the completion of training, using “classifier.predict,” a forecast of the outcome of the test set has been made after the training is complete. To evaluate the models, the evaluation metrics: confusion matrix, accuracy, sensitivity, and F1 score were calculated for all models as indicated in Table 1.

VI. MODEL EVALUATION AND RESULTS

TABLE 1: Evaluation Metrics

ML Model	Accuracy (%)	Sensitivity	F1 Score
kNN	90.3	1.0	0.903
DTC	92.5	1.0	0.975
LR	81.5	1.0	0.815
SVM	84.3	1.0	0.843
RFC	94.1	1.0	0.940

The Random Forest Classifier (RFC) has demonstrated exceptional performance in the conducted experiments, surpassing the other tested algorithms. It achieved an impressive accuracy rate of 94.1%, indicating its ability to effectively distinguish between malware and benign instances within the "Edge-IIoTset" dataset. This high accuracy rate is a testament to the robustness and effectiveness of the RFC algorithm in detecting and classifying malicious activities within IoT systems.

Moreover, sensitivity, which measures the RFC's ability to correctly identify positive instances (i.e., correctly identifying malware), was found to be 1.0. This indicates that the RFC algorithm successfully detects all instances of malware in the dataset without missing any. This high sensitivity is particularly crucial in the context of security operations, as it ensures that potential threats are not overlooked or mistakenly classified as benign.

Additionally, the RFC algorithm achieved an impressive F1 score of 94, indicating a well-balanced performance in terms of precision and recall. The F1 score is a measure of a model's accuracy, considering both false positives and false negatives, and its high value demonstrates the RFC's ability to strike a good balance between accurately detecting malware instances and minimizing misclassifications.

Consequently, the superior performance of the RFC algorithm in terms of accuracy, sensitivity, and F1 score showcases its effectiveness in malware detection for IoT systems. This highlights its potential as a valuable tool for security operation analysts in identifying and mitigating threats effectively, thus enhancing system security and reducing the risk of unauthorized access and data breaches.

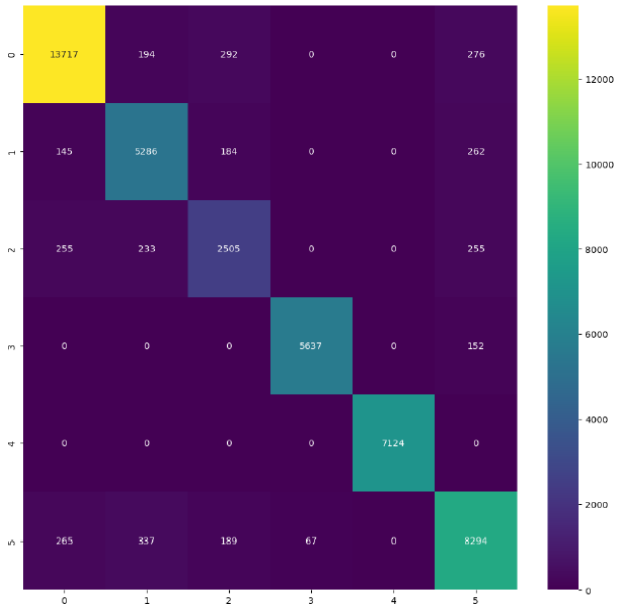

Figure 2: Confusion matrix for RFC

VII. CONCLUSIONS

The rising complexity and volume of cyber-attacks necessitate proactive measures to enhance system security and prevent unauthorized access and data breaches. AI-based learning models enable security operation analysts to gain a deeper understanding of the nature of cyber threats and devise more effective strategies for mitigating them. This study specifically focused on the pre-processing, analysis, and evaluation of data collected from digital sensors within an IoT system to identify potential vulnerabilities associatedwith IoT and IIoT networking protocols. To accomplish this,various Machine Learning (ML) algorithms, including k- Nearest Neighbors (KNN), Decision Tree Classification Algorithm (DTC), Logistic Regression (LR), Support VectorMachines (SVM), and Random Forest Classifier (RFC), were assessed using the publicly available "Edge-IIoTset" dataset for malware detection. The experimental results clearly demonstrate that the Random Forest Classifier (RFC) outperformed the other algorithms, achieving an impressive 94% accuracy in malware detection. This highlights the significant impact of AI in strengthening system security and protecting against malicious attacks in IoT environments. Further, this work has limitations. This work is specifically designed for a particular sensor dataset, and as a result, the performance of algorithms and models may vary depending on the specific IoT dataset being tested.

VIII. FUTURE RESEARCH DIRECTIONS

The future research would be on investigating the efficacy of transfer learning techniques in the context of malware detection. By leveraging pre-trained models on large-scale datasets, researchers can explore how to transfer knowledge and adapt deep learning models to effectively detect

malware in IIoT networks. Further,developing robust deep learning models that are resilient to adversarial attacks in IIoT networks. This involves studying different attack vectors and devising defense mechanisms to enhance the security and reliability of the models.

REFERENCES

[1] H. Hassani, E. S. Silva, S. Unger, M. TajMazinani, and S. Mac Feely, "Artificial Intelligence (AI) or Intelligence Augmentation (IA): What Is the Future?," *AI*, vol. 1, no. 2, Art. no. 2, Jun. 2020, doi: 10.3390/ai1020008.

[2] J. C. Flowers, "Strong and Weak AI: Deweyan Considerations".

[3] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," in *2018 Fourth International Conference on Computing Communication Control and Automation(ICCUBEA)*, Aug. 2018, pp. 1–6. doi 10.1109/ICCUBEA.2018.8697857.

[4] P. Ongsulee, V. Chotchaung, E. Bamrungsi, and T. Rodcheewit, "Big Data, Predictive Analytics and Machine Learning," in *2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE)*, Nov. 2018, pp. 1–6. doi: 10.1109/ICTKE.2018.8612393.

[5] R. Kakadiya, R. Lemos, S. Mangalan, M. Pillai, and S. Nikam, "AI Based Automatic Robbery/Theft Detection using Smart Surveillance in Banks," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Jun. 2019, pp. 201–204. doi: 10.1109/ICECA.2019.8822186.

[6] T. Kaluarachchi, A. Reis, and S. Nanayakkara, "A Review of Recent Deep Learning Approaches in Human-Centered Machine Learning," *Sensors*, vol.21, no. 7, Art. no. 7, Jan. 2021, doi: 10.3390/s21072514.

[7] S. Dilek, H. Çakır, and M. Aydın, "Applications of Artificial Intelligence Techniques to Combating Cyber Crimes: A Review," *IJAIA*, vol. 6, no. 1, pp. 21–39, Jan. 2015, doi: 10.5121/ijaia.2015.6102.

[8] Z. Allam and Z. A. Dhunny, "On big data, artificial intelligence and smart cities," *Cities*, vol. 89, pp. 80–91, Jun. 2019, doi: 10.1016/j.cities.2019.01.032.

[9] T. Ban, N. Samuel, T. Takahashi, and D. Inoue, "Combat Security Alert Fatigue with AI-Assisted Techniques," in *Cyber Security Experimentation and Test Workshop*, in CSET '21. New York, NY, USA: Association for Computing Machinery, Sep. 2021, pp. 9–16. doi: 10.1145/3474718.3474723.

[10] Z. Ullah, F. Al-Turjman, L. Mostarda, and R.Gagliardi, "Applications of Artificial Intelligence and Machine learning in smart cities," *Computer Communications*, vol. 154, pp. 313–323, Mar. 2020, doi: 10.1016/j.comcom.2020.02.069.

[11] M. Migdal, "Council Post: The Vertical Approach: Why AI Should Be Industry Specific," *Forbes*. https://www.forbes.com/sites/forbestechcouncil/2021/05/28/the-vertical-approach-why-ai-should-be-industry-specific/ (accessed May 21, 2023).

[12] M. Kuzlu, C. Fair, and O. Guler, "Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity," *Discov Internet Things*, vol. 1, no. 1, p. 7, Feb. 2021, doi: 10.1007/s43926-020-00001-4.

[13] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020, doi: 10.1109/COMST.2020.2986444.

[14] A. F. Jahwar and S. R. M. Zeebaree, "A State of the Art Survey of Machine Learning Algorithms for IoT Security," *Asian Journal of Research in Computer Science*, vol. 9, no. 4, Art. no. 4, Jun. 2021, doi: 10.9734/ajrcos/2021/v9i430226.

[15] "Edge-IIoTset Cyber Security Dataset of IoT & IIoT." https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot (accessed May 21, 2023).

[16] "Information | Free Full-Text | Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence." https://www.mdpi.com/2078-2489/11/4/193 (accessed May 21, 2023).

[17] "K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint," *www.javatpoint.com*. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning (accessed May 17, 2023).

[18] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. v. n. Santhosh Kumar, M. Selvi, and K. Arputharaj, "Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks," *IET Communications*, vol. 14, no. 5, pp. 888–895, 2020, doi: 10.1049/iet-com.2019.0172.

[19] "Decision Tree Algorithm in Machine Learning - Javatpoint." https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (accessed May 17, 2023).

[20] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augment Hum Res*, vol. 5, no. 1, p. 12, Mar. 2020, doi: 10.1007/s41133-020-00032-0.

[21] "Logistic Regression: Equation, Assumptions, Types, and Best Practices." https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/ (accessed May 17, 2023).

[22] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.

[23] "What is a Support Vector Machine? - Datatron." https://datatron.com/what-is-a-support-vector-machine/ (accessed May 17, 2023).