Intrusion Detection in Smart Homes Using K-Nearest Neighbors and Decision Trees Algorithm on IoT Network Traffic for Attack Classification

Andrea Menjívar, Master en IoT[1], Jose Luis Ordoñez-Avila, Phd en Dirección Empresarial[2], Manuel Cardona, Phd en Robótica[3]
[1]Florida International University, Florida, USA
[2]Facultad de Ingeniería, Universidad Tecnológica Centroamericana (UNITEC), San Pedro Sula, Honduras,
[3]Dirección de Investigación, Universidad Don Bosco, San Salvador, El Salvador
anmenj003@fiu.edu, jlordonez@unitec.edu, manuel.cardona@udb.edu.sv

Abstract*– Many homes now feature smart technology and numerous devices connected to the Internet, exposing them to cyberattacks. Therefore, implementing protection mechanisms to identify, predict, and mitigate these threats to smart home devices is crucial. This research proposes two machine learning models—K-Nearest Neighbors and Decision Tree—to predict malicious activity in smart home connections and classify whether an attack is occurring. The study presents both models along with an in-depth analysis of their performance, assessing how they function on unseen data and their effectiveness on the dataset. The findings highlight the strengths and weaknesses of each model, providing valuable insights into their applicability in real-world scenarios. By offering a comparative evaluation, this research contributes to the ongoing efforts in enhancing the security of smart homes and underscores the importance of adopting advanced machine learning techniques for intrusion detection systems (IDS). This study aims to lay the groundwork for future developments in smart home cybersecurity solutions.

Keywords-IDS, IoT, Smart Home, K-Nearest Neighbors, Decision Tree, Network Security.

I. Introduction

In recent years, the use of the Internet of Things (IoT) has grown exponentially and rapidly, driven by significant technological advancements and the increasing demand for a more interconnected world [1]. Smart homes represent a significant application of the Internet of Things (IoT). It refers to the technology that connects various sensors and actuators through a communication network or protocol, allowing daily tasks to be performed conveniently while enhancing comfort in houses. This integration not only streamlines household activities but also enables users to monitor and control their home environment remotely [2]. The global smart home market was valued at USD 45.8 billion in 2017 and is projected to reach USD 114 billion by the end of 2025 [3].

Forgetting to turn off the lights, TV, kitchen oven, or lock the door when leaving home is no longer an issue, thanks to smart home IoT technology. The market for smart home devices is projected to grow at a compound annual rate of 16.9% [4]. However, as smart home systems continue to advance, they have also become prime targets for cybercriminals seeking to gain control of IoT devices and access sensitive information. New threats and security concerns are emerging, since many of these devices have weak security systems that can be easily exploited by these cybercriminals [5]. Criminals often attempt unauthorized access to smart homes by exploiting vulnerabilities such as weak authentication, unencrypted communication between devices, lack of verification, password cracking, malware, and Denial-of-Service (DoS) attacks [6, 7]. As smart homes continue to gain popularity, it is crucial to develop practical security solutions that can protect residential customers while accommodating the unique characteristics of smart home environments [8]. To address these challenges, researchers have surveyed existing smart home safety and security systems, to understand their architecture, enabling technologies, and components.

II. Literature Review

A. Machine Learning

Another popular piece of technology that has risen in the past few years is Machine Learning. Machine learning, ML, refers to algorithms that allow systems to learn from data and improve their performance over time without being explicitly programmed. Machine Learning is a branch of Artificial Intelligence that allows computers to learn from data and make accurate predictions based on past experiences. It encompasses a variety of techniques that analyze input data according to predefined rules to create effective prediction models. ML can identify patterns and make informed decisions, which is particularly valuable in addressing security concerns in IoT smart homes [9, 10].

In smart homes, ML is used to enhance security by analyzing patterns in network traffic, detecting anomalies, and predicting potential threats. By learning from historical data, these systems can identify unusual behaviors indicative of security breaches, such as unauthorized access or device tampering, thus enabling proactive measures against cyber threats [11]. For instance, indicators of compromise (IOCs) can be used to develop datasets on Internet of Things (IoT) threats to train machine learning models, aiding in the real-time monitoring, detection, and response to threats. Additionally, ML classification algorithms can leverage IOC data to categorize malware behavior effectively [12]. As

shown in Figure 1 Machine Learning Models can be trained to identify normal traffic, to detect attacks and to predict whether an attack is about to happen.

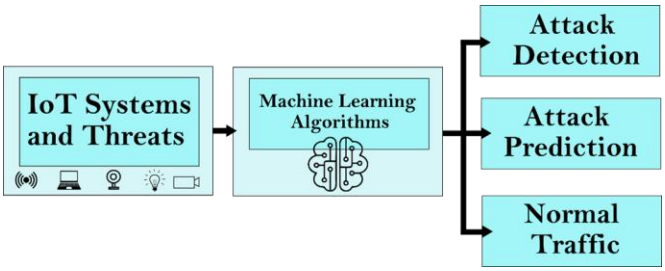

Fig. 1. IoT Systems and Threats

B. K-Nearest Neighbor and Decision Trees Algorithms

Various ML algorithms, including decision trees, K-nearest neighbors, and multi-layer perceptron's, have shown high accuracy in predicting user actions, with reported accuracy exceeding 97%. [13]. These advancements in ML-powered smart homes aim to improve quality of life while reducing energy consumption by minimizing human intervention and security issues. The kNearest Neighbor (KNN) algorithm is a simple, yet effective machine learning technique widely used for classification and regression tasks. It classifies new data points based on their similarity to previously labeled data, assigning them to the class of their nearest neighbors. For this model, the data set provided is labeled and the data points are categorized into two or more classes, so that model can predict the class of the unlabeled data. The algorithm's performance is influenced by key parameters such as the number of neighbors (k), distance function, and weighting function [14].

The Decision Tree algorithm is a supervised learning method used for classification and regression tasks [15]. These algorithms create tree-like structures where internal nodes represent attribute tests, branches denote test outcomes, and leaf nodes indicate classes. Classification is performed by traversing the tree from root to leaf based on attribute tests [16]. Decision trees have emerged as a valuable tool for enhancing smart home functionality. They can be used to detect unusual activities by analyzing sensor data [17], recognize human activities with high accuracy [18], and predict inhabitant behaviors based on appliance usage patterns [19].

This literature review examines the existing research on machine learning applications for enhancing security in smart homes, focusing specifically on K-Nearest Neighbors (KNN) and Decision Tree algorithms. Various studies have demonstrated the effectiveness of these techniques in detecting and preventing intrusions, addressing the unique challenges posed by IoT environments. Authors in [20] suggest an architecture based on SDN to enhance the security of smart home networks. This approach utilizes K-Nearest Neighbor (KNN) for classifying devices and detecting malicious traffic. In [21] a anomaly detection model based on MahalaNobis distance calculated by KNN models. [22], [23], [24] also applied KNearest Neighbor algorithm to smart home related projects. Decision Trees algorithm is also widely used, for example [25], proposed a decision tree model constructed from automation rules that control the operations of IoT devices in a smart home to detect anomalies. In [26] authors also used Decision Trees algorithm to predict smart home lightning behavior. [27], [28], and [29] also applied decision tree for their smart home related research.

The K-Nearest Neighbors (KNN) and Decision Tree algorithms are increasingly utilized to address security challenges in smart homes, providing effective solutions for intrusion detection and threat assessment. These techniques leverage their ability to analyze patterns in data and make informed predictions, enhancing the overall security of IoT devices. In the following section, a literature review will explore various studies conducted by other authors on this topic.

III. METHODS

. The rise of smart home technologies has introduced significant security vulnerabilities in IoT devices. As more households adopt these systems, the risk of cyberattacks escalates. For this project, a machine learning model was developed, combining both K-Nearest Neighbors (KNN) and Decision Tree algorithms. This model analyzes the rich network traffic data sent by IoT devices, enabling it to identify whether the data represents a cyberattack or not.

The K-Nearest Neighbors (K-NN) and Decision Tree models were selected because they offer a balance between accuracy, interpretability, and ease of implementation, making them well-suited for smart home environments. K-NN detects anomalous behavior by comparing network patterns with previous instances, without requiring complex assumptions about the data. Meanwhile, Decision Trees produce clear and understandable rules, enabling quick attack identification with transparent reasoning. Both models perform well with moderate-sized datasets and adapt effectively to the dynamic and diverse nature of traffic in connected devices, making them ideal for real-time intrusion detection systems.

The dataset used for this project was the Smart Home Intrusion Detection Dataset available at Kaggle [30], which is an online platform that contains several datasets and a collaborative environment to build and share machine learning models and solutions. This dataset contains various features, ranging from basic network metrics to complex interaction patterns, making it ideal for classifying normal and potentially malicious operations. It captures network traffic data from IoT devices in a smart home, with features related to connection metrics and interaction patterns. The primary goal of the presented model is to use this data to predict and detect cyber intrusions in real-time by distinguishing between normal network activity and potentially malicious behavior.

Some key features that this dataset contains include: the duration of the connection, the protocol used, type of network service, as well as SRC and DST bytes which is the volume of the data being sent from source to destination and vice versa. SRC bytes can aid in the detection of large-scale data transfers or probing activities as for DST bytes they can potentially indicate responses to request or data exfiltration attempts. The data set also contains other relevant features such as land, wrong fragment, urgent, logged in user, number or compromised conditions, number of connections to the same service and rate of connections that resulted in SYN errors, among others. The last column of the data set, which is the target column indicates whether the connection was part of an attack yes or no, serving as the label for the model training. This model combines both K-Nearest Neighbor algorithms to classify network traffic into clusters representing normal and anomalous behaviors. Using KNN to group similar traffic patterns, helping identify deviations that could indicate intrusions. The Decision Tree was built to establish rules for identifying intrusions based on the network traffic features, providing a clear structure for detecting malicious activity

A. Data Preprocessing

To start the building of the model, all the necessary libraries were imported, and the data set was loaded. Once loaded the shape of the data set was obtained which had a total of 148,517 rows with 24 columns meaning that it had 23 different features and the last column being the target column. Next, the dataset was examined to see the different data types for each column. This step was crucial for ensuring that the features were in the correct format for analysis and modeling. Ensuring proper data types facilitates further processing steps in the model.

When using the KNN algorithm, it's essential to ensure that all input features are numeric and appropriately scaled. All the numeric columns were normalized using the MinMaxScaler to ensure they were all on the same scale to avoid feature domination. As for the categorical columns, they were converted into a numeric format using One-Hot Encoding. After the data preprocessing the dataset ended up with 102 columns and was ready to be trained.

Before training the model, some analysis was made of the data to get more information on what the model will be working on and to have a visual understating of each attack type and features. Figure 2 shows the total number of attacks classified as Yes or No. This figure shows that the dataset for this model is balanced meaning that the distribution of the two classes is even, which is beneficial for model training. This will help avoid bias toward one class, so it helped the model make more accurate predictions.

B. Modeling with KNN and Decision Tree

After preprocessing the data was split into training and testing, using 80% of the data for training and 20% for testing. This step is also crucial for building a machine learning model, model to learned patterns from the training data while reserving the unseen testing data for evaluating the performance of the model. The KNN algorithm will use the testing data to classify whether an IoT network interaction is an attack or not, based on the proximity of the data points in the feature space. For this model a K=3 was considered meaning the model considers the 3 nearest neighbors to classify data points. After the KNN model was trained, the next step involved training a Decision Tree model. The model learned by recursively splitting the dataset based on feature values, allowing IoT network traffic to specific attack category. The decision tree helps provide insight into how different sensor data features contribute to determining whether an interaction is a potential cyberattack or normal activity.

After training both models, the next step was making predictions. Each model (KNN and Decision Tree) applied its learned patterns to classify the test dataset, determining whether an interaction was a potential cyberattack or normal activity. The predictions represent the model's classification of IoT interactions based on the test features. This process is crucial for assessing how well each model generalizes to unseen data, providing insight into their ability to detect potential security threats in new IoT network traffic.

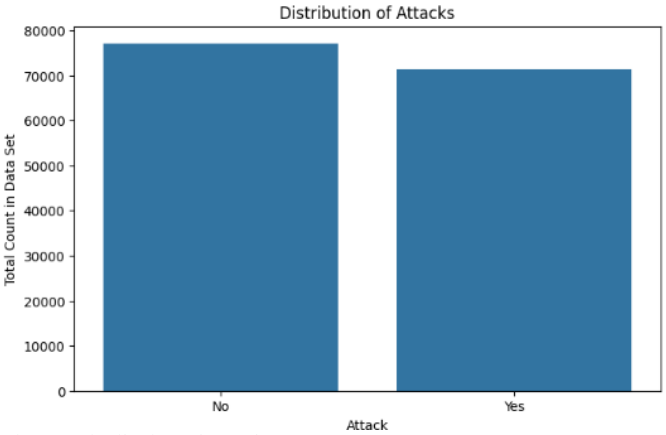

Fig. 2. Distribution of Attacks

III. RESULTS

After the predictions were made, the next step involved the evaluation of both models. The classification reports for both the K-Nearest Neighbors (KNN) and Decision Tree models indicate a strong performance in identifying IoT interactions as either attacks or normal activities. Both models achieved a remarkable accuracy of 99%, with identical precision, recall, and F1-scores for both classes (0.0 and 1.0). This consistency suggests that the models are highly effective in distinguishing between the two classes, minimizing false positives and negatives. The high scores across all metrics demonstrate the robustness of the models, making them suitable for real-time applications in cybersecurity, where accuracy is paramount. However, further validation on

additional datasets would be beneficial to assess their generalization capabilities in diverse scenarios.

This analysis focuses on the ten most important features identified in the dataset, which are crucial for understanding the underlying patterns in IoT interactions. This chart enhances interpretability and provides insights into how these features may influence predictions. Figure 3 illustrates that one of the most critical features in determining whether an interaction is classified as an attack is the Flag SF feature. The ten most importante feutres are described here:

1. **flag_SF**
   - Indicates the final status of a network connection.
   - "SF" means "Successful connection with no errors".
   - High importance: Normal traffic usually ends in SF, whereas many attacks do not.
2. **dst_host_diff_srv_rate**
   - Percentage of connections to the same destination host using different services.
   - Useful for detecting port scans.
3. **service_eco_i**
   - Indicates if the connection uses the eco_i service.
   - Certain attacks may target or use specific services.
4. **service_ecr_i**
   - Indicates if the connection uses the ecr_i service.
5. **hot**
   - Number of "hot" indicators in the connection, such as access to system files or suspicious commands.
   - Useful for identifying potentially malicious activity.
6. **service_private**
   - Indicates if the service used in the connection is labeled as "private".
   - Private services are often associated with malicious behavior or attacks.
7. **dst_host_count**
   - Number of connections to the same destination host within a time window.
   - High values can indicate a DDoS attack or scanning behavior.
8. **dst_bytes**
   - Number of bytes sent from the destination host (server) to the source (client).
   - Abnormal values might signal data exfiltration or irregular behavior.
9. **dst_host_same_srv_rate**
   - Percentage of connections to the same host using the same service.
   - Low rate may point to scanning activity.
10. **dst_host_srv_count**
    - Number of connections to the same host and service.
    - Repeated patterns can indicate abnormal behavior.

This flag represents the connection status and is vital for understanding the expected behavior of network communications. In legitimate traffic, the Flag SF indicates a successful connection establishment, while anomalies in this flag's patterns, such as an unexpected frequency of connections or rapid state changes, can signal malicious activities. This insight is crucial for developing effective intrusion detection systems that can promptly identify and respond to cyber threats. Other important features include the Echo and Echo Response service. Analyzing these services can help identify legitimate versus potentially malicious network activity, as certain attacks may attempt to exploit uncommon or unauthorized services. Understanding these services enhances the model's ability to detect anomalies and improves overall network security.

Figure 4 shows a correlation heatmap of the 10 most important features to visually demonstrate the relationships between these key variables and their influence on attack detection. This heatmap provides an intuitive understanding of how these features interact, revealing potential patterns or redundancies. By focusing on the most significant features, it allows for a more concise analysis of their combined effect on model performance, ultimately aiding in the refinement of the model and enhancing interpretability for stakeholders in the context of network security.

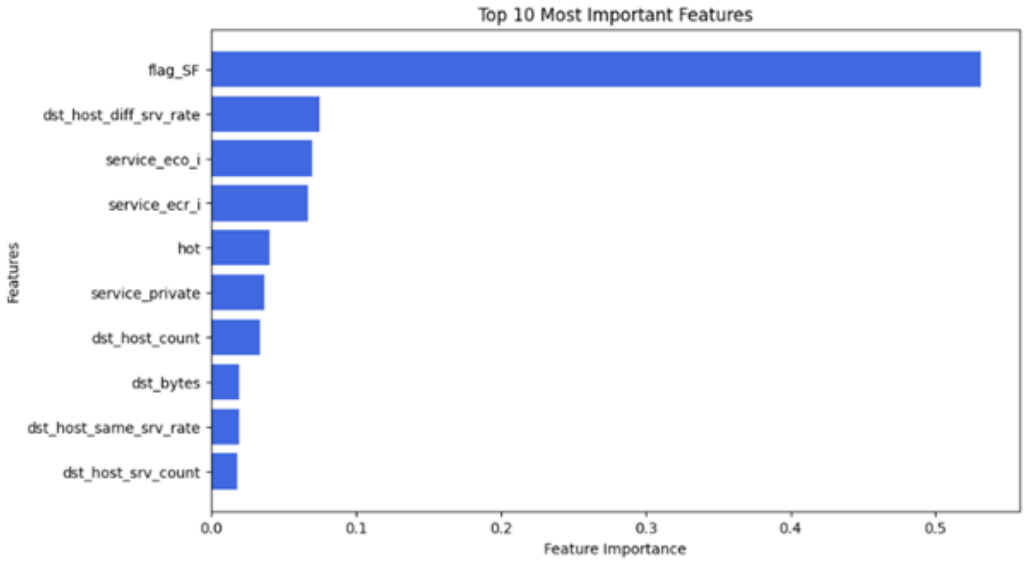

Fig. 3. Top 10 Most Important Feature

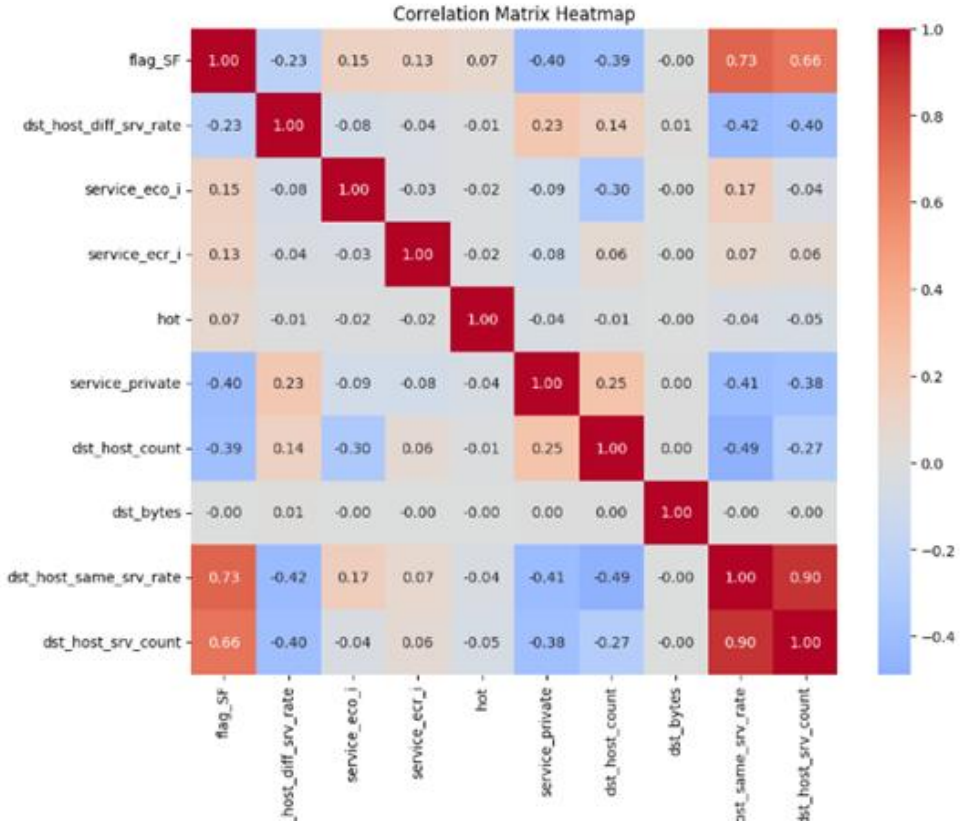

Fig. 4. Correlation Heatmap

A score list was added for KNN model with values of k from 1 to 5 to examine how variying the number of nearest neighbors affects the model's accuracy. This comparison helps identify the optimal *k*-value for this specific dataset, ensuring that the KNN model isn't overfitting or underfitting. By testing different k-values it was determined the point where the model performs best, balancing simplicity and accuracy, which is critical in detecting activities or potential intrusions in the IoT network traffic analysis. Figure 5 shows the graph of the results of this score list.

Based on this accuracy results for different values of k in KNN model, it is evident that accuracy initially decreases as *k* increases from 1 to 5. The highest accuracy of 0.998768

occurs at k=1, indicating that the model is highly sensitive to the training data and may be overfitting.

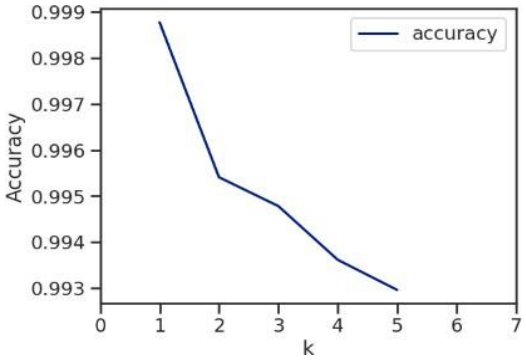

Fig. 5. Scorelist K Values

A. *Cross validation*

In this section of the analysis, cross-validation was employed to assess the robustness and reliability of the KNN and Decision Tree models. By using 5-fold cross-validation, the data was split into five subsets, allowing each model to be trained and validated on different portions of the dataset. This approach helps to mitigate overfitting and ensures that the models generalize well to unseen data. Figure 6 shows the results of the cross-validation for both models. The resulting cross-validation scores provide insight into each model's performance consistency, highlighting the Decision Tree's slight edge over KNN in classification accuracy.

Based on the cross-validation scores, both the KNN and Decision Tree models exhibit high performance, with mean scores around 0.99. The Decision Tree slightly outperforms KNN, suggesting it may generalize better to unseen data. The consistent scores across folds indicate stability in the models' predictions, reinforcing their reliability for distinguishing between normal activity and potential cyberattacks in IoT interactions.

B. *Validation scores*

Based on the accuracy results for both the KNN and Decision Tree models, the KNN model demonstrated a high training accuracy of approximately 99.85% and a validation accuracy of about 99.12%. In contrast, the Decision Tree model had a training accuracy of around 99.44% and a validation accuracy of 98.99%. The KNN model's higher validation accuracy suggests it is better at generalizing unseen data compared to the Decision Tree, which may indicate that it captures the underlying patterns in the data more effectively. However, the slight difference in performance highlights the importance of tuning model parameters and understanding the trade-offs involved in each approach.

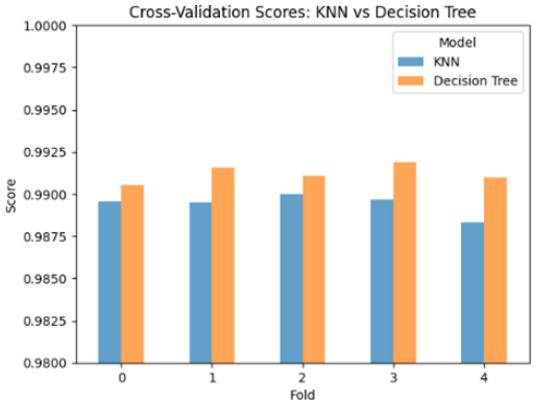

Fig. 6. Cross-Validation Score

IV. CONCLUSION

After training and validating the effectiveness of these two models K-Nearest Neighbors and Decision Tree algorithms for detecting intrusion activities within a smart home IoT Network, the models were able to accurately classify activities, demonstrating the applicability of machine learning techniques in IoT security systems. This research project underlines the potential for AI in smart homes to enhance protection against evolving cyber threats. Both models showed strong performance in classifying IoT interactions based on the data set that was used. The classification report provides a detailed evaluation of the performance for both KNN and Decision Tree models. KNN shows a higher precision, recall and F1-scores for certain key classes, which indicates its ability to deliver accurate and comprehensive classifications. Especially for these IoT activities. Meanwhile, the Decision Tree's performance is comparable, but with slightly lower generalization to unseen data. While both models exhibit reliable cross-validation scores, the KNN model's stronger results on unseen data make it more robust for real-world applications, emphasizing the importance of tuning to enhance overall performance. This type of work should be regulated as part of a comprehensive policy [31], especially for medical IoT applications [32-34].

Future work could expand on these findings by incorporating more complex datasets, exploring additional algorithms, or integrating real-time traffic monitoring. Such advancements could significantly improve the scalability and effectiveness of intrusion detection in smart homes, ensuring a more comprehensive defense mechanism against potential threats.

REFERENCES

[1] A. G. Menjivar, "Mobile App Development for Wireless Monitoring and Configuration of Sensors using ESP8266," in 2022 IEEE Central America and Panama Student Conference (CONESCAPAN), Oct. 2022, pp. 1–6. doi: 10.1109/CONESCAPAN56456.2022.9959569.

[2] H. Ahmad, M. Gupta, and Shivam, "A Study on Implementation of Cyber Physical Systems in Smart Home," in 2023 IEEE 11th Region 10 Humanitarian Technology Conference (R10-HTC), Oct. 2023, pp. 58–63. doi: 10.1109/R10-HTC57504.2023.10461741.

[3] A. A. Saleem, M. M. Hassan, and I. A. Ali, "Smart Homes Powered by Machine Learning: A Review," in 2022 International Conference on Computer Science and Software Engineering (CSASE), Mar. 2022, pp. 355–361. doi: 10.1109/CSASE51777.2022.9759682.
[4] D. S. Nagarkar, D. P. Mishra, and D. V. Gaikwad, "Factors At Play: Investigating The Dimensions Of Privacy And Security In Smart Home Environments," Educ. Adm. Theory Pract., vol. 30, no. 5, Art. no. 5, Apr. 2024, doi: 10.53555/kuey.v30i5.3414.
[5] H. F. Al-Turkistani and N. K. AlSa'awi, "Poster: Combination of Blockchains to Secure Smart Home Internet of Things," in 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Nov. 2020, pp. 261–262. doi: 10.1109/SMARTTECH49988.2020.00069.
[6] "Trend Micro Security Predictions for 2019".
[7] W. Ali, G. Dustgeer, M. Awais, and M. A. Shah, "IoT based smart home: Security challenges, security requirements and solutions," in 2017 23rd International Conference on Automation and Computing (ICAC), Sep. 2017, pp. 1–6. doi: 10.23919/IConAC.2017.8082057.
[8] C. Lee, L. Zappaterra, K. Choi, and H.-A. Choi, "Securing smart home: Technologies, security challenges, and security requirements," in 2014 IEEE Conference on Communications and Network Security, Oct. 2014, pp. 67–72. doi: 10.1109/CNS.2014.6997467.
[9] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and Md. S. H. Sunny, "Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review," IEEE Access, vol. 7, pp. 13960–13988, 2019, doi: 10.1109/ACCESS.2019.2894819.
[10] A. A. Saleem, M. M. Hassan, and I. A. Ali, "Smart Homes Powered by Machine Learning: A Review," in 2022 International Conference on Computer Science and Software Engineering (CSASE), Mar. 2022, pp. 355–361. doi: 10.1109/CSASE51777.2022.9759682.
[11] M. M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," Computers, vol. 12, no. 5, Art. no. 5, May 2023, doi: 10.3390/computers12050091.
[12] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions," Electronics, vol. 9, no. 7, Art. no. 7, Jul. 2020, doi: 10.3390/electronics9071177.
[13] A. A. Saleem, M. M. Hassan, and I. A. Ali, "INTELLIGENT HOME: EMPOWERING SMART HOME WITH MACHINE LEARNING FOR USER ACTION PREDICTION," Sci. J. Univ. Zakho, vol. 11, no. 3, Art. no. 3, Aug. 2023, doi: 10.25271/sjuoz.2023.11.3.1145.
[14] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.
[15] I. Rahmatillah, E. Astuty, and I. D. Sudirman, "An Improved Decision Tree Model for Forecasting Consumer Decision in a Medium Groceries Store," in 2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS), Aug. 2023, pp. 245–250. doi: 10.1109/ICIIS58898.2023.10253592.
[16] Q. Dai, C. Zhang, and H. Wu, "Research of Decision Tree Classification Algorithm in Data Mining," Int. J. Database Theory Appl., vol. 9, no. 5, pp. 1–8, May 2016, doi: 10.14257/ijdta.2016.9.5.01.
[17] V. Stankovski and J. Trnkoczy, "Application of Decision Trees to Smart Homes," in Designing Smart Homes: The Role of Artificial Intelligence, J. C. Augusto and C. D. Nugent, Eds., Berlin, Heidelberg: Springer, 2006, pp. 132–145. doi: 10.1007/117884858.
[18] "Decision Trees for Human Activity Recognition Modelling in Smart House Environments." Accessed: Oct. 04, 2024. [Online]. Available: https://www.sne-journal.org/sne-volumes/volume-28/sne284-articles/decision-trees-for-human-activity-recognition-modelling-insmart-house-environments/
[19] M. Marufuzzaman, T. Tumbraegel, L. F. Rahman, and L. M. Sidek, "A machine learning approach to predict the activity of smart home inhabitant," J. Ambient Intell. Smart Environ., vol. 13, no. 4, pp. 271–283, Jan. 2021, doi: 10.3233/AIS-210604.
[20] H. Gordon, C. Park, B. Tushir, Y. Liu, and B. Dezfouli, "An Efficient SDN Architecture for Smart Home Security Accelerated by FPGA," in 2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Jul. 2021, pp. 1–3. doi: 10.1109/LANMAN52105.2021.9478836.
[21] H. M. Ngo, D. H. Ha, Q. D. Pham, T. V. Le, and S. H. Nguyen, "Detecting Anomaly in Smart Homes Based on Mahalanobis Distance," in ICC 2024 IEEE International Conference on Communications, Jun. 2024, pp. 2204–2209. doi: 10.1109/ICC51166.2024.10622234.
[22] S. Kang and J. W. Yoon, "Classification of home appliance by using Probabilistic KNN with sensor data," in 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Dec. 2016, pp. 1–5. doi: 10.1109/APSIPA.2016.7820745.
[23] M. I. Siddiq, I. P. D. Wibawa, and M. Kallista, "Integrated Internet of Things (IoT) technology device on smart home system with human posture recognition using kNN method," IOP Conf. Ser. Mater. Sci. Eng., vol. 1098, no. 4, p. 042065, Mar. 2021, doi: 10.1088/1757899X/1098/4/042065.
[24] Z. Wang, J. Gao, R. Chen, and J. Wang, "A Modified KNN Algorithm for Activity Recognition in Smart Home," ICEB 2018 Proc. Guilin China, Dec. 2018, [Online]. Available: https://aisel.aisnet.org/iceb2018/96
[25] M. Wang, C. Fu, and X. Du, "Decision-Tree Based Root Cause Localization for Anomalies in Smart IoT Systems," in ICC 2021 IEEE International Conference on Communications, Jun. 2021, pp. 1–5. doi: 10.1109/ICC42927.2021.9500348.
[26] I. B. P. P. Dinata and B. Hardian, "Predicting smart home lighting behavior from sensors and user input using very fast decision tree with Kernel Density Estimation and improved Laplace correction," in 2014 International Conference on Advanced Computer Science and Information System, Oct. 2014, pp. 171–175. doi: 10.1109/ICACSIS.2014.7065885.
[27] V. G. Sanchez and N.-O. Skeie, "Decision Trees for Human Activity Recognition in Smart House Environments," 222-229, 2018, doi: 10.3384/ecp18153222.
[28] "Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid — IEEE Journals & Magazine — IEEE Xplore." Accessed: Oct. 04, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7434588
[29] J. Maitre, G. Glon, S. Gaboury, B. Bouchard, and A. Bouzouane, "Efficient Appliances Recognition in Smart Homes Based on Active and Reactive Power, Fast Fourier Transform and Decision Trees".
[30] "Smart Home Intrusion Detection Dataset." Accessed: Oct. 04, 2024. [Online]. Available: https://www.kaggle.com/datasets/bobaaayoung/dataset-invade
[31] K. A. Castro, J. A. Siwady, E. Castillo, A. Alonzo, M. Cardona and M. E. Perdomo, "Artificial Intelligence for All: Challenges and Harnessing Opportunities in AI Democratization," *2024 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, San Salvador, El Salvador, 2024, pp. 143-148, doi: 10.1109/ICMLANT63295.2024.00030.
[32] J. L. Ordoñez-Ávila, F. Danilo Bonilla and E. D. García, "Development of a teleoperated robot based on an experimental design for sterilization in Honduras," 2021 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT), Soyapango, El Salvador, 2021, pp. 1-6, doi: 10.1109/ICMLANT53170.2021.9690534.
[33] J. L. Ordoñez-Avila, M. Cardona, D. A. Aguilar, M. Ordoñez and C. L. Garzón-Castro, "A Novel Monitoring System for Contagious Diseases of Patients using a Parallel Planar Robot," *2022 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, Soyapango, El Salvador, 2022, pp. 1-6, doi: 10.1109/ICMLANT56191.2022.9996485.
[34] Tagrid Gabrie, Alberto Max Carrasco, and Jose Luis Ordoñez Avila. 2021. Low-cost Robot Assistance Design for Health Area to Help Prevent COVID-19 in Honduras. In Proceedings of the 6th International Conference on Robotics and Artificial Intelligence (ICRAI '20). Association for Computing Machinery, New York, NY, USA, 283–288. https://doi.org/10.1145/3449301.3449349