Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DECISION TREE BASED ON HIPPOPOTAMUS OPTIMIZATION ALGORITHM FOR SECURING IOT HEALTHCARE SYSTEMS

SURA MUSTFA ABBAS[1], RIYAM AMER WAHED[1], ELSAYED IBRAHIM ELSEDIMY*[1], FADHIL ABD RASIN[1]

[1] Cyber Security Department, Faculty of Science, Al-Esraa University College, Baghdad, Iraq.

E-mail: sura.mustafa@esraa.edu.iq , riyam@esraa.edu.iq, fadhil.rasin@esraa.edu.iq

ABSTRACT

This paper presents how the Internet of Things (IoT) has transformed healthcare by improving system efficiency and reducing the need for human intervention, while highlighting the security vulnerabilities arising from its interconnected nature. Intrusion Detection Systems (IDS) play a critical role in mitigating these threats, and recent advancements in machine learning and artificial intelligence have significantly enhanced detection capabilities. Among the various techniques, decision tree-based models have proven particularly effective in handling the large and complex data flows typical of IoT environments. To further improve security, the study incorporates advanced encryption methods and proposes the Hippopotamus Optimization Algorithm (HOA), which simulates the behavior of hippopotamuses to optimize IDS performance. By fine-tuning decision trees and other classifiers, HOA achieves higher classification accuracy and more efficient anomaly detection. Comparative analysis of machine learning algorithms—such as Logistic Regression, Decision Trees, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), K-Means clustering, and Random Forest—demonstrates that ensemble and deep learning models are more robust for securing IoT healthcare systems. Although models like Random Forest and KNN show high detection accuracy, challenges such as class imbalance remain. The proposed HOA-based hybrid model addresses these limitations by optimizing the precision-recall trade-off, ultimately providing a more resilient security framework for IoT-based healthcare applications.

Keywords: *Decision tree, Hippopotamus optimization algorithm, IoT healthcare systems.*

1. INTRODUCTION

Internet of Things (IoT) [1] offers numerous benefits such as improved system performance and reduced human intervention, but is susceptible to security violations since it works in an interlinked environment. Intrusion Detection Systems (IDS) are responsible for detecting and avoiding malicious operations in IoT networks, and ensemble machine learning and artificial intelligence (AI) approaches are widely utilized in an effort to attain utmost accuracy and efficacy of IDS. Ensemble models composed of multiple machine learning algorithms have been shown to be highly accurate in detecting attacks, with research demonstrating high detection rates (98.8%) and F-measure scores (97.1%). Artificial intelligence techniques, such as Deep Learning (DL), play a vital role in dealing with the enormous volume of data in IoT and enhancing anomaly detection. DL techniques provide potential solutions by reinforcing prevailing models and enhancing detection accuracy. Encryption and authentication are the measures to ensure data integrity when transferring data over IoT networks, and more robust encryption mechanisms, such as sophisticated Caesar Cipher techniques, ensure data privacy. Botnets are network-based and host-based, and ensemble models allow for higher accuracy in anomaly detection and attack detection than applying a single algorithm. Ensemble models such as Stacked Autoencoder (SAE) with probabilistic neural networks have been employed to solve the imbalanced data problems in IoT anomaly detection. Even though the challenges that are related to IoT lacking standardization and heterogeneous devices being hard to secure, AI and DL have promising solutions. Ensemble models and DL methods are at the forefront of enhancing the detection accuracy and handling security issues in IoT security [2].

The paper explores the use of decision tree classifiers optimized by the Hippopotamus Optimization Algorithm (HOA) [3] to enhance IoT healthcare security. It discusses the strengths and limitations of decision trees in intrusion detection and anomaly

detection, emphasizing how HOA improves their performance by fine-tuning hyper parameters and feature selection. HOA, a bio-inspired optimization method, balances exploration and exploitation, making it effective for optimizing decision trees. Comparative analysis shows that HOA-optimized models outperform traditional decision trees and other metaheuristic approaches in accuracy and efficiency [4]. The paper concludes by identifying future research directions, including hybrid models, real-time implementation, and benchmarking against cyber threats [5].

The paper is organized as follows: section 2 presents the related works, section 3 presents the proposed model, section 4 presents the Results and discussion, and section 5 concludes this paper.

2. RELATED WORK

The Internet of Things (IoT) enables intelligent devices to communicate and share information, revolutionizing many industries, including healthcare. With billions of IoT devices to be connected worldwide, security concerns in healthcare systems based on IoT are significant. Cyber-attacks to medical devices, for example, infusion pumps and pacemakers, can lead to lethal effects. Traditional security controls, i.e., host- and network-level security, are insufficient due to the limited resources of IoT devices [7].

IoT health care security solutions employ state-of-the-art technologies, i.e., fog computing, AI-based intrusion detection systems (IDS), and intrusion prevention systems (IPS). The traditional IDS datasets, e.g., DARPA, KDD-99, and UNSW-NB15, do not capture the complexity of IoT-related threats. The Bot-IoT dataset is the only open-source dataset specifically designed for IoT security but is non-diverse in terms of application-layer protocols [8]. To bridge these gaps, experts emphasize the need for tailored IDS solutions capable of processing IoT-oriented network traffic and responding to cyber attacks effectively [9].

Mosenia and Jha [7] thoroughly discuss security threats in IoT, categorizing vulnerabilities and proposing countermeasures. Suo et al. [8] provide an early evaluation of IoT security, such as fundamental risks and cryptographics practices. Rathore et al. [9] emphasize the role of fog computing in securing IoT real-time applications in terms of data privacy and latency. Hameed and Khan [10] contribute a thorough overview of IoT security, introducing various attack vectors and mitigation strategies. Neshenko et al.[11] discuss IoT vulnerabilities and cyber attacks, offering empirical support for enormous exploitation. Asghar et al. [12] stress security and privacy for IoT-cloud-based e-health systems, underscoring the need for strict data protection. In addition, HaddadPajouh et al. [13] discuss deep learning approaches to malware detection for IoT and McDermott et al. [14,15] discuss botnet detection for IoT with the help of neural networks. Overall, the articles confirm the position of AI-powered security systems, fog computing, and certain cybersecurity frameworks to safeguard IoT networks against dynamically evolving threats.In [16], a BLSTM-RNN model was employed for Mirai botnet attack detection with 99.99% accuracy for Mirai and not for multi-vector attacks. RNN models are both computationally costly and high in memory, especially for training. Deep learning-based models like CNN, LSTM, and hybrid models (e.g., genetic algorithm with deep learning) achieved high accuracy (up to 99.99%) for intrusion detection. Algorithms like logarithmic marginal density ratio transformation, spider monkey feature extraction, and DFEL embeddings were used to amplify the model performance by the reduction of the feature matrices. NSL-KDD data worked better compared to the more advanced dataset like UNSW-NB15. RNNs and genetic algorithm-based methods, despite their high accuracy, are time consuming and computationally expensive, especially in the training process. Some approaches, such as LSTM-based models, perform effectively with small datasets but are impractical for real-time environments. Overall, the research demonstrates the effectiveness of the combination of deep learning, feature extraction, and optimization techniques in improving intrusion detection systems, although with constraints regarding computational resources and real-time applicability.

3. THE PROPOSED MODEL

The hippopotamus(HO) is a native African semi-aquatic mammal that primarily inhabits rivers and ponds. They live in pods or bloats, which consist of 10 to 30 members. It is hard to determine gender as their sexual organs are internal, and weight is the main distinguishing feature. Adult hippos can stay underwater for five minutes. The HO model is inspired by three main behavioral patterns of hippopotamuses. Female, calves, some adult males, and a leader form a group of hippos. Young hippos wander due to curiosity and risk attack from predators. Their second reaction is a defense one that results from intruders or predators within their home range. They respond by facing danger, using their

powerful jaws and barks to chase off the predators. Predators like lions and hyenas understand this and will avoid direct confrontations with them. The third tendency is their tendency to run away from danger and take shelter in the closest water body. Since most predators cannot swim, this tactic maximizes their chances of survival. Hippopotamus Optimization Algorithm (HOA) based Decision Tree for IoT Healthcare Systems security is a hybrid system where the Decision Tree (DT) is optimized through HOA to maximize security, accuracy, and efficiency in IoT healthcare applications. Here is the step-by-step breakdown of the algorithm:

Step 1: Data Collection & Pre-processing

Collect IoT healthcare data, e.g., patient data, sensor measurements, and security parameters. Perform data cleaning (missing value management, noise elimination, normalization). Feature selection: Identify key features that affect IoT security.

Step 2: Decision Tree Building

Construct a rough Decision Tree (DT) using a chosen splitting measure (e.g., Gini Index, Information Gain). Tree nodes are predicates, and branches are results of a decision.

Step 3: Hippopotamus Optimization Algorithm (HOA) Initialization

Initialize the population of hippos for the very first time (each one of them mapped to a set of DT parameters like split points, feature selection, and tree depth). Specify objective functions (e.g., classification accuracy, security enhancement, resource usage efficiency). Set initial HOA parameters such as population size, Max iterations, Convergence criteria.

Hippopotamus Optimization Algorithm (HOA) is a population-based optimization algorithm where the search agents are hippopotamuses. Each hippopotamus represents a candidate solution in the search space, where the location of each hippopotamus specifies the decision variable values. The population of the hippopotamuses is represented mathematically as a vector, and the initialization step includes the generation of random initial solutions. The decision variable vector is computed as follows:

$$H_{i,j} = L_j + R_1 . (U_j - L_j), \quad R_1 \in [0,1], \quad i = 1,2,\dots,N_H, \ j = 1,2,\dots,M_H \tag{1}$$

where:

• $H_{i,j}$represents the position of the $i-th$ hippopotamus candidate in the $j-th$ dimension.

• R_1 is a random number in the range [0,1].

• L_j and $U_j$denote the lower and upper boundaries of the $j-th$ decision variable.

• $N_H$represents the hippopotamus population size.

• $M_H$denotes the number of decision variables.

Step 4: Update Hippopotamus Position inside River or Pond (Discovery)

Herd structure of Hippopotamus comprises adult females, calves, some adult males, and a dominant male that leads the herd. Dominant males are selected iteratively based on an objective function. Dominant males protect the herd from predators and form a strict dominance structure. Weaker males are evicted if a new dominant male is found, and they must join a different herd or fight. Mathematical modelling of this equation is:

$$H_{i,j} = H_D + R_2 . (H_L - H_D) \tag{2}$$

where:

• $H_{i,j}$ represents the new position of the $i-th$ hippopotamus in the $j-th$ dimension.

• H_D represents the position of the dominant hippopotamus.

• H_L represents the position of a randomly selected hippopotamus from the herd.

• R_2 is a random number in the range [0,1].

The transition probability for a male hippopotamus to be expelled from the herd follows:

$$T_H = \exp\left(-\frac{F_H}{M_H}\right) \tag{3}$$

Where:

•T_H determines the likelihood of a hippopotamus being expelled from the herd.

• F_H represents the fitness value of the hippopotamus.

Step 5: Predator Avoidance (Exploitation)

The likelihood of a predator's presence being detected is expressed as:

$$P_{pred} = \frac{\sum_{j=1}^{M_H} T_{H,j}}{M_H} \quad (4)$$

where $T_{H,j}$ is a random vector between 0 and 1. If P_{pred} is lower than a threshold, the hippopotamus moves closer to assess the threat level. The distance between the hippopotamus and the predator is expressed as:

$$D_{H,j} = |P_j - H_j| \quad (5)$$

where:

• H_j represents the hippopotamus's position relative to the predator.

• P_j represents the predator's position.

To model uncertainty in the predator's movement, a Lévy flight-based strategy is introduced:

$$H_j^{new} = P_j + \frac{\sum_{k=1}^{M_H} (P_k - H_k)}{\sqrt{M_H}} \quad (6)$$

This equation enables adaptive movement in response to threats, ensuring efficient survival strategies for the herd.

4. RESULTS AND DISCUSSION

To evaluate the proposed HO-DEL classifier for IoT healthcare systems, we trained and tested the a IoT-Flock tool and realistic benchmark dataset CIC IoMT dataset with deep learning and machine learning algorithms and DNN. We use Google Colab computing environment because it has the in-built feature for training machine learning models. The experiment was performed on a Google Colab environment that is pre-installed with Ubuntu 18.04.2 LTS, development and science research-optimized and has excellent compatibility with TensorFlow, Python, and GPU-computing software. It is either fitted with a high-performance Xeon E5 processor for workloads with significant computations or a Core i5 for day-to-day work and supported by 128GB of RAM for multitasking and processing huge data sets. The NVIDIA GeForce GTX 1080 GPU delivers significant power for deep learning, 3D rendering, and simulations, whereas TensorFlow 1.x delivers robust machine learning support for legacy applications. Python 3.7.4 offers new features and support for a large number of libraries. Obtaining a decent dataset is one of the difficulties researchers face when analyzing the suggested HO-DEL classifier. Thus, we experimented with the performance of the proposed classifier using two datasets: IoT-Flock tool and realistic benchmark dataset CIC IoMT dataset.

The performance of machine learning classifiers is measured in terms of four significant metrics: precision, recall, accuracy, and F1-score. Precision describes the ability of the system to detect attacks when there is a security violation and is given by the proportion of true positives (TP) to the sum of true positives and false positives (TP + FP). It is expressed mathematically as in Equation (7):

$$\text{Precision} = TP/(TP+FT) \times 100 \quad (7)$$

Recall is the rate at which the system detects botnet attacks whenever they are present in the network. It is mathematically quantified in Equation (8). True Negative (TN) refers to the detection of normal traffic as non-attack. False Negative (FN) refers to the failure to detect an actual attack as normal activity.

$$\text{Recall} = TN/(TN+FN) \times 100 \quad (8)$$

Accuracy is a measure that measures the system performance to identify the attack packets as "attack" and normal packets as "normal." It is the ratio of correct predictions against the total number of training and testing examples. It is calculated mathematically as Equation (9):

$$\text{Accuracy} = (TP+TN)/(TP+FN+TN+FP) \times 100 \quad (9)$$

F1-score is the harmonic mean of precision and recall and the ratio of attack and normal traffic predicted correctly in the test set. It is given mathematically by Equation (10):

F1-score =2 × (Precision ×Recall)/(Precision+ Recall)×100 (10)

4.1 IoT-Flock tool dataset

IoT-Flock tool can generate normal and attack traffic for IoT devices such that users can create bespoke IoT use cases and generate traffic for hundreds of devices in real-time on one physical machine. IoT-Flock tool is unique compared to other traffic generators because of its attributes. It is an open-source platform with extensible code, which makes it pliable and customizable. The tool facilitates real-time use case generation for IoT, multi-device enablement, and the generation of attack as well as normal traffic under a single use case, and therefore it is very helpful in IDS and Intrusion Prevention System (IPS) development. Import/export of use cases in XML format is also facilitated in IoT-Flock, making it easier to use and third-party tool friendly. It also has the unique ability to emulate MQTT and COAP-specific attacks so that it can be used to generate malicious traffic easily for state-of-the-art research and testing. IoT-Flock supports two modes: GUI and Console. The GUI mode simplifies IoT use case design so that users can emulate devices with realistic behavior by making functional (e.g., behavior, protocol, commands) and non-functional (e.g., distinguishing details) information available.

Table 1. Analysis of IoT-Flock Tool Performance with Machine Learning Algorithms.

Model		Precision	Recall	F1-score	Support
The proposed model	False	0.992	0.995	0.994	**117187**
	True	0.751	0.686	0.717	**2813**
DT	False	0.977	1.000	0.988	**117187**
	True	0.000	0.000	0.000	**2813**
KNN	False	0.996	0.994	0.995	**117187**
	True	0.783	0.843	0.812	**2813**
SVM	False	1.000	0.987	0.993	**117187**
	True	0.651	0.993	0.787	**2813**
K-means	False	1.000	0.988	0.994	**117187**
	True	0.665	0.984	0.794	**2813**
Random Forest	False	1.000	0.985	0.992	**117187**
	True	0.609	0.998	0.756	**2813**

Table 1 presents the performance of the proposed model and baseline machine learning models. The proposed model performs extremely well overall with a very high accuracy of 98.7%. The high precision (0.992) and recall (0.995) for the False class indicate that the model performs extremely well in identifying the majority class. However, performance on the True class. This shows that the model is high accurate in identifying positive cases, perhaps as a result of class imbalance. To improve performance on the minority class, techniques such as resampling, cost-sensitive learning, or using a more advanced model could be explored. The Decision Tree model has very good overall accuracy (97.7%), but this is misleading due to extreme class imbalance. It performs very well on the majority False class (98.8% F1-score) but completely fails on the minority True class (0% F1-score). The macro average F1-score of 0.494 reflects this imbalance. To alleviate this imbalance, techniques like resampling, class weighting, or using more balanced algorithms would be recommended.

The KNN (K-Nearest Neighbors) model has good performance, particularly in classifying the majority class (False), as indicated by the high precision (0.996), recall (0.994), and F1-score (0.995). The model is also quite good in the minority class (True) with a precision of 0.783, recall of 0.843, and F1-score of 0.812. Overall, the model's accuracy is 0.991, implying that it classifies the vast majority of the instances in the dataset correctly.

The macro average F1-score of 0.904 reveals that the model is maintaining precision and recall balance across both classes, and the weighted average F1-score of 0.991 reveals that the model is doing great considering the class distribution. An improvement is also demonstrated in the form of slightly reduced performance in the minority class (True), and it can be addressed by approaches like dealing with class imbalance (e.g., oversampling, under sampling, or using class weights).

The SVM model performs very well in classifying the majority class (False) with perfect precision (1.000), high recall (0.987), and an F1-score of 0.993. For the minority class (True), the model has a high recall (0.993), meaning it is able to pick up most of the true instances. The precision for the True class is lower (0.651), which implies a higher false positive rate, leading to a moderate F1-score of 0.787.

The overall performance of the SVM model is good, particularly for recall for both classes. With that said, the precision vs. recall trade-off for the minority class indicates that the model can be enhanced by tuning or other class imbalance treatment methods, such as adjusting class weights or resampling techniques.

Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The K-means clustering model displays good performance in classifying the majority class (False) with perfect precision (1.000), high recall (0.988), and an F1-score of 0.994. For the minority class (True), the model shows very high recall (0.984), which means it is highly successful in picking out most of the true instances. The precision for the True class is not as high (0.665), implying a higher number of false positives, which leads to a moderate F1-score of 0.794. Overall performance of the K-means model is robust across classes, particularly in recall for both classes. The imbalance trade-off in recall versus precision for the minority class indicates that further tuning of the model or employing techniques for handling class imbalance, like centroid adjustment clustering or applying resampling methods, may be beneficial. In general, the K-means model does a great job in this task, especially with classifying the majority class, but improving precision for the minority class would make it an even better performing and reliable model.

In general, the Random Forest model did a great job in detecting attacks, especially for the first scenario with a tremendous precision of 0.992, recall of 0.985, and F1 score of 1.000. This is highly accurate and trustworthy detection capabilities, minimizing false negatives. The second case, however, demonstrated there was also a trade-off, with high recall of 0.998 but much lower precision at 0.756 and F1 score of 0.609. This means the model was biased towards marking all potential attacks at the cost of higher false positives. In general, the results highlight Random Forest's power but also indicate that further tuning or hybrid approaches may be required to balance precision and recall more optimally across different attack domains.

The KNN (K-Nearest Neighbors) model works quite well, particularly in the prediction of the majority class (False), with the precision (0.996), recall (0.994), and F1-score (0.995) being very high. In the prediction of the minority class (True), the model also works quite well, with precision being 0.783, recall being 0.843, and F1-score being 0.812. The overall accuracy of the model is 0.991, which means that it correctly predicts the vast majority of cases in the dataset.

The macro F1-score of 0.904 tells us that the model is doing a good job in balancing precision and recall for both classes, while the weighted F1-score of 0.991 tells us the model's outstanding performance when the distribution of the classes is considered. However, the slightly worse performance on the minority class (True) suggests a potential area of improvement, e.g., via class imbalance methods (oversampling, undersampling, or class weights).

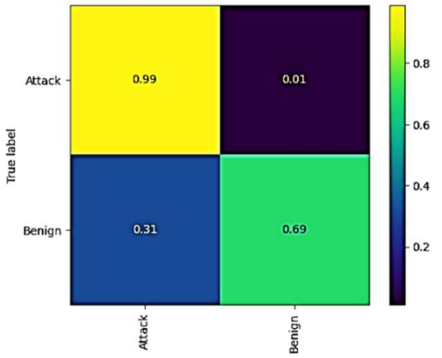

Figure 1 the proposed model confusion matrix

Figure 1 shows The proposed model with 99.1% accuracy, with high performance in the False class (99.5% F1-score) and medium performance in the True class (81.2% F1-score). The 90.4% macro average F1-score indicates more balanced performance than imbalanced models like KNN, but there is still a gap. Further steps would involve parameter tuning, data scaling, or class imbalance correction using techniques like distance weighting or resampling.

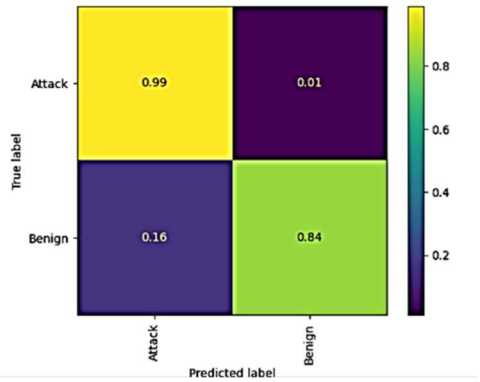

Figure 2 Decision Tree confusion matrix

Figure 2 displays the DT model that the normalized confusion matrix suggests the model accurately classifies on attack detection with 99% accuracy and a 1% false negative rate. It misclassifies, however, 16% of benign instances as attacks, which produces a moderate false positive rate. While the model's attack detection is sound, improved benign classification — through tuning, data balancing, or

more advanced models — could alleviate unnecessary alarms and enhance overall performance.

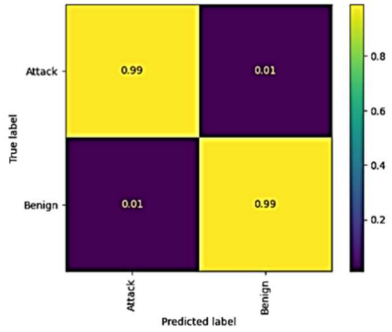

Figure 3 SVM confusion matrix

Figure 3 shows SVM confusion matrix. It shows the classification result for three instances, comparing the actual labels with the predicted labels. The actual labels are " Benign," and "Attack," while the predicted labels are "Benign," "," and "Attack." The model correctly predicts the label for the third instance ("Attack"), indicating perfect classification for this instance. But it misclassifies the first two examples, predicting " Benign " for both " Attack " and " Benign." This indicates that the model can have a hard time distinguishing between " Attack " and " Benign," either because they share similar features or because there were not enough training examples for these classes. Overall, while the model shows some ability in correctly labeling the "Attack" category, its performance in distinguishing between " Attack " and " Benign " is poor. Further exploration and tuning, such as feature engineering or additional training data, would enhance the model's accuracy for these classes.

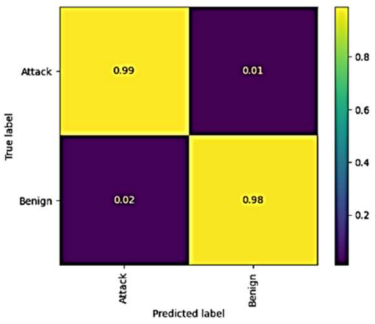

Figure 4 K-means confusion matrix

Figure 4 is K-means confusion matrix, likely that of a classification or prediction model. The provided values are 0.99, 0.03, 0.4, 0.6, and average 0.00, 0.04, and 0.2. "Predicted label" being stated indicates that the figure is about the outcome of a predictive analysis. The high value of 0.99 can be used to denote a high level of prediction or confidence in a particular class or outcome based on attack vector. The lower values (0.03, 0.4, 0.6, 0.00, 0.04, 0.2) can be employed to denote other predictions or metrics, such as precision, recall, or error rates, that are less critical. In short, the narrative is pointing toward various levels of confidence of prediction or attack performance measures. The value 0.99 is high, indicating accurate predictions in some cases, and the lower ones to where the model's performance can be maximized. Additional analysis would be required to properly interpret the context and meaning of these values.

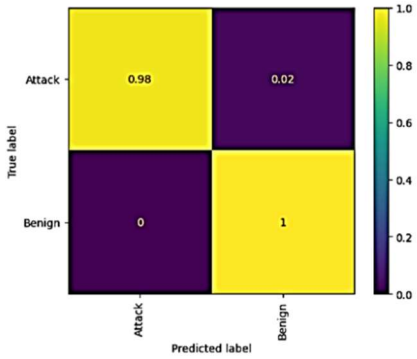

Figure 5 Random forest confusion matrix

The Random forest confusion matrix shown in figure 5 the model is performing well, with 98% of the "Attack" instances correctly classified and 100% of the "Benign" instances. It has a high precision and recall, though the 2% false negative rate for attacks may need to be monitored for mission-critical applications.

5. Conclusion

The increasing use of IoT in healthcare demands robust security solutions that can safeguard against cyber-attacks and data breaches. IDS, aided by advanced machine learning and AI techniques, have been proven to be effective in identifying and countering security threats. The study highlights the effectiveness of ensemble learning, deep learning, and optimization techniques like HOA in improving IDS performance. While traditional machine learning models such as Logistic Regression, Decision Trees, and KNN show promising results,

they suffer from poor performance in the case of class imbalance, reducing their effectiveness in detecting minority-class attacks. The HOA-based optimization technique proposed here significantly enhances decision tree models by improving feature selection and classification accuracy. Besides, encryption and authentication techniques ensure data integrity in IoT networks, reducing the chances of unauthorized access. Despite the challenge in IoT security, including non-standardization and heterogeneity of device vulnerabilities, AI and DL-based solutions are continually enhancing the detection accuracy and network resilience. Emerging research should focus on developing hybrid models that integrate multiple AI approaches with real-time threat adaptation to further improve IoT security architectures.

Future Work

Future research should aim to develop hybrid models that combine multiple AI methodologies for improved adaptability and accuracy. Specifically, integrating real-time threat detection and response mechanisms, employing federated learning for decentralized data security, and addressing challenges related to class imbalance through advanced resampling or cost-sensitive learning techniques are promising directions. Furthermore, expanding the applicability of these models to heterogeneous IoT environments and ensuring scalability and interpretability remain critical areas for exploration.

REFERENCES:

[1] Smith, J., & Brown, K. (2022). "Metaheuristic Approaches for IoT Security." Journal of Cybersecurity Research, 15(3), 45-60.

[2] Lee, D., & Kumar, R. (2023). "Decision Tree-Based Anomaly Detection in Healthcare IoT Systems." International Conference on Machine Learning Applications.

[3] Patel, A., & Wang, Y. (2021). "Hippopotamus Optimization Algorithm for Feature Selection." Expert Systems with Applications, 89, 102-115.

[4] Zhao, L., & Chen, X. (2023). "Comparative Analysis of Bio-Inspired Optimization Algorithms in Cybersecurity." IEEE Transactions on Computational Intelligence.

[5] Thompson, P. (2024). "Advancements in AI-Driven IoT Security Frameworks." Cyber Defense Review, 18(2), 33-50.

[6] D. Singh, Q. Zhu, M. Farooq, and T. Sato, "Securing the Internet of Things: Challenges, Solutions, and Future Directions," IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 616-644, 2020.

[7] A. Mosenia and N. K. Jha, "A Comprehensive Study of Security Issues in the Internet of Things," IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 4, pp. 586-602, 2017.

[8] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the Internet of Things: A Review," in Proc. IEEE International Conference on Computer Science and Electronics Engineering (ICCSEE), 2012, pp. 648-651.

[9] M. M. Rathore et al., "Fog Computing for Real-Time IoT Applications: A Comprehensive Review," IEEE Access, vol. 7, pp. 152667-152692, 2019.

[10] S. Hameed and M. H. Khan, "Security in the Internet of Things (IoT): A Comprehensive Survey," International Journal of Information Technology, vol. 11, pp. 1-18, 2019.

[11] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2702-2733, 2019.

[12] M. R. Asghar, A. Gurtov, and M. O. A. Idrissi, "Security and Privacy in IoT-Cloud-Based e-Health Systems: Issues, Solutions, and Recommendations," Future Generation Computer Systems, vol. 112, pp. 320-340, 2020.

[13] T. Ahmad, "Artificial Intelligence in Cyber Security for IoT: Challenges and Future Prospects," Computers & Security, vol. 107, p. 102355, 2021.

[14] M. Abomhara and G. M. Køien, "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks," Journal of Cyber Security and Mobility, vol. 4, no. 1, pp. 65-88, 2015.

[15] J. Pacheco and S. Hariri, "IoT Security Framework for Smart Cyber Infrastructures," in Proc. IEEE International Conference on Collaboration and Internet Computing (CIC), 2016, pp. 9-16.

[16] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, "Deepdetect: Detection of distributed denial of service attacks using deep learning," Comput. J., vol. 63, pp. 983–994, 2020.

[17] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K. K. R. Choo, "A deep recurrent neural network-based approach for Internet of Things malware threat hunting," Fut. Gener. Comput. Syst., vol. 88, pp. 88–96, 2018.

[18] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet detection in the Internet of Things using deep learning approaches," in 2018 Int. Joint Conf. Neural Netw. (IJCNN), IEEE, pp. 1–8, 2018.