Detection and Classification of Malicious Software in the IoT Environment Using Feature Selection by Deep Learning Methods

Saad Talib Hasson
University of Babylon
Babylon, Iraq
saad.aljebori@uobabylon.edu.ig

Murtdha Saadoon Balasim
Imam Ja'afar Al-Sadiq University
Baghdad, Iraq
Murtaza.saadoun@ijsu.edu.iq

Mohammed Shakir Mohmood
Scholarship & Cultural Relations Directorate, Ministry of Higher Education & Scientific Research
Baghdad, Iraq
Mahmood@tut.by

***Abstract*— In the rapidly evolving landscape of Internet of Things (IoT) security, the detection and classification of malware present significant challenges. This study delves into the efficacy of various machine learning and deep learning models in classifying IoT malware. Primarily focusing on a Convolutional Neural Network (CNN) enhanced by Gray Wolf Optimization (GWO), the research demonstrates the model's superior accuracy, precision, recall, and F1 score in malware identification. The dataset used, IoT-23, encompasses 23 distinct IoT malware families, providing a comprehensive basis for model training and evaluation. Comparative analysis with other machine learning models like Random Forest, SVM, XGBoost, and deep learning architectures like LSTM, GRU, DenseNet, and InceptionV3 reveals the nuanced capabilities of each in handling IoT malware classification. This study's findings highlight the critical role of model selection in IoT cybersecurity, emphasizing the need for tailored solutions based on specific dataset characteristics and computational constraints. The results underscore the potential of CNN+GWO as a leading approach in the ongoing battle against IoT malware threats**

Keywords*— Deep Learning Methods, IoT, Classification Algorithms, Machine Learning

I. Introduction

The increasing number of IoT devices, which will be online by 2025, presents a significant threat to users. These devices can be accessed from anywhere, exposing them to threats such as unauthorized access to personal information and security vulnerabilities. Intrusion detection systems (IDSs) are crucial in protecting IoT networks against intrusions. However, due to the limited bandwidth, energy, memory, and CPU capabilities of IoT devices, complex IDSs are needed. Denial of service (DoS) attacks are severe and devastating, causing financial losses for companies and organizations [1–5]. Attackers use IoT devices' flaws to perform denial-of-service attacks, making protection a top priority for researchers worldwide. IDSs are categorized by their detection abilities: signature, specification, or anomaly-based [6–8]. Signature-based IDSs identify attacks when a device or network connection compares with a signature in the IDS database, while anomaly-based IDSs alert when a behavior profile deviates beyond a predetermined threshold [9–11]. Specification-based IDSs detect intrusions when network behavior deviates from standards, but manual criteria have fewer false positives than anomaly-based criteria and do not require training.

Global trade increases the need for protected information in the workplace and daily life, as every global structure uses computer networks. Network security is essential for securing sensitive data, and identifying network breaches is crucial for protecting sensitive data. An intrusion detection model based on feature selection can be used to identify assaults on structures and enhance intrusion detection using collected data [12–15]. Integrating malicious packets to a user's system or poor configuration can lead to intelligent intrusions, which can combine multiple vulnerabilities in a global network. Effective intrusion detection is essential for protecting sensitive data and ensuring the safety of IoT devices. Fig. 1 shows the IDS Basic Structure.

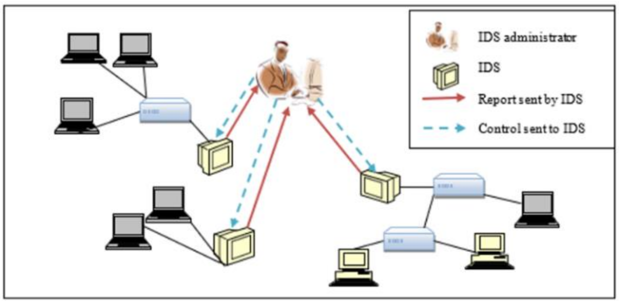

Fig. 1. IDS Basic Structure

A. Types of Intrusion Detection System

- Network intrusion detection systems (NIDS) are designed to monitor all network traffic and match it to known attacks. They are installed at a designated point within the network, observing all devices and matching traffic to known attacks. If an attack or abnormal behavior is detected, an alert is sent to the administrator [21]. Fig. 2 shows the NIDS Architecture.

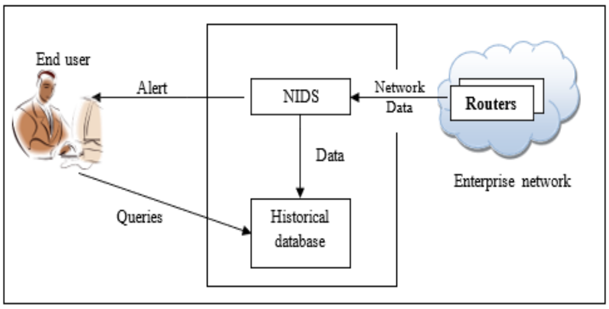

Fig. 2. NIDS Architecture

- Host Intrusion Detection Systems (HIDS) are independent network devices that monitor incoming and outgoing packets and alert administrators if suspicious activity is detected. They compare system files and send alerts if they are edited or deleted. HIDS is particularly useful on mission-critical machines that don't change their layout [21]. Fig. 3 represented the HIDS Architecture.

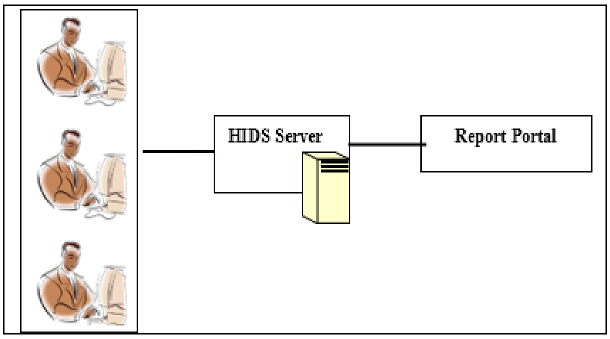

Fig. 3. HIDS Architecture

B. IDS Architecture and Attacks

IOT architecture consists of four layers where each layer has is specific function. IoT architecture diagram is given below with explanation. Fig. 4 represented the IOT Architecture.

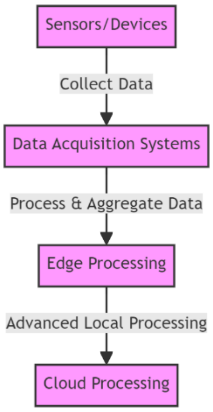

Fig. 4. IOT Architecture

C. Analysis Approach

The two main categories through which network can be analyzed for the detection of intrusion are:

Misuse detection: It is an approach that uses paradigm matching to identify intrusions. It establishes aberrant structural behavior by collecting the assault paradigm, and normal behavior is characterized by matching it against previously recorded attacks. This approach challenges deviation apprehension access, which defines typical system behavior while labeling abnormal activity.

- However, effective intrusion detection requires updating the database of assaults.

- Anomaly detection: An Anomaly-Based Intrusion Detection System is a framework that is used to determine computer intrusions and abuse by monitoring system activity and classifying it as either normal or abnormal. This is accomplished via the use of anomaly detection. The classification relies on rules as opposed to paradigms or signatures, and it is designed to detect any kind of inappropriate usage that deviates from the typical operation of the system [22, 23].

D. Basic Categories of Intrusions

The four categories in one of which the simulated attacks fall in:

- Denial of Service (DoS) attack: It is an attack where an attacker makes a computer resource unavailable to legitimate users by making it too busy or full to handle valid requests. This attack can be carried out without requiring a user or attacker to log in, as repeated requests cause the host to become too busy [24].

- User-to-root (U2R): These attacks involve attackers gaining access to an operating system as a regular user and exploiting a vulnerability to access the system's root. These content-based attacks target users, with buffer overflow attacks being the most common type. NSL-KDD (7) identifies file creations and shell prompts as key characteristics.

- Remote to Local Attacks (R2L): These occur when hackers transmit packets without an account on a system, gaining local access as if they were a user. Recognizing R2L assaults is challenging due to network and host-level characteristics. The NSL-KDD data set helps identify R2L attacks by considering duration, service requested, and number of unsuccessful login attempts.

- Probing Attack: This kind of attack tries to obtain knowledge about a computer network in order to take control of the network's security. A probing attack was launched in order to gather all of the necessary information on the target, which established the groundwork for more destructive assaults. In order to detect a probe attack, "duration of connection" and "source bytes" are the two primary characteristics. A broad assault is one that is often investigated.

E. Challenges

IoT expansion raises data security worries. No consistent methodology exists to verify recommended systems. The study effort reveals the estimate of their approaches in IoT systems that rely on their implemented dataset, as well as one unique problem that does not work on real statistics and other concerns. It's hard to design an IDS that's adaptable, deployable, online, and flexible for all stakeholders. Most written on this subject relies on created datasets, contains part or all of the technique, and shows findings using skewed criteria. This article reviews contemporary IoT intrusion detection difficulties. Developing a real-time IoT anomaly detection system is complex. This form of IDS must first understand routine activity to predict aberrant or suspicious behavior. No external assaults or attack traffic are ensured during learning. Without these fixes, this IDS will create many false alerts. Data pre-processing, feature reduction, model formulation and execution, and ML-based IDS techniques increase computational overhead. Building a low-computing-demand IDS is another challenge. More threat detection research is needed to prevent future attacks, and security weaknesses like confidentiality and privacy must be rectified and avoided [26–30]. From Table 1 show IoT Dataset Used For IDS.

TABLE I. IoT DATASET USED FOR IDS

DATASETS	MERITS	DEMERITS
KDDCUP99 [25]	Labeled data may be found in this dataset. In addition to the class label, each connection is evaluated using 41 different attributes. The KDD99 classification techniques are imbalanced. There are no new assaults in the dataset.	The KDD99 classification algorithms are skewed. The dataset contains no new assaults.
UNSW-NB15 [25]	Create CSV files for network traffic. It includes nine distinct attack types: analysis, fuzzers, dos, backdoors, reconnaissance, worms, exploit, shell, and generic.	It is more complicated than that of the KDD99 dataset due to similar behaviour of recent attacks and typical network traffic.
NSL-KDD [25]	KDDCUP99 is superior. Overcome the restrictions of KDDCUP99	Inadequate contemporary attack

II. RELATED WORK

This section reviews previous research on enhancing attack detection and game theory concepts in Wireless Sensor Networks (WSNs). It consists of four sections: 1) on clustering and cluster head selection, 2) on IoT and game theory models for IDS framework development, and 3) on various attacks detection techniques and algorithms. The chapter analyzes the state of the art in WSN security and existing techniques for reliability, accuracy, and high attacks detection rates. It is divided into different areas for a broader perspective [29–31].

A. *Cluster Head Selection*

During the clustering process, sensor nodes are grouped together, and one node is chosen as the Cluster Head (CH) for that group. The CH acts as a data collector, relaying information from other nodes to the BS. Methods of clustering may extend the life of networks, lessen their energy footprint, and make them scalable. Data gathering, central hub (CH) selection, hierarchical routing, data aggregation, and fusion are the four main components of clustering protocols. Algorithms for Cluster Head Selection that are gentle on energy use can extend the life of networks. While current CH selection techniques take into account the residual energy of sensor nodes to maximize efficiency, they sacrifice throughput in the process. Research challenges in Wireless Sensor Networks often center on the process of clustering and choosing a cluster leader.

B. *Review on Swarm Intelligence Algorithm*

Damien Wohwe Sambo et al. (2019) found that centralized cluster solutions based on the Swarm Intelligence paradigm are more suitable for applications with low power consumption, high data rates, or high scalability than other algorithms. Weifeng Sun et al. (2020) analyzed a representative Swarm Intelligence algorithm and their IoT applications, focusing on SI-enabled applications in wireless sensor networks (WSNs) and related WSN research issues. Li Cao and colleagues (2017) reviewed swarm intelligence optimization algorithms and key technologies used in mobile wireless sensor networks, including MWSNs. They discussed the concept, classification, and architecture of the Internet of Things and MWSN, and the latest results of swarm intelligence algorithms for optimizing MWSN performance.

C. *Implementation of PSO in Cluster Head Selection*

Based on criteria including residual energy, cluster distances, and distances between sensor nodes, Aparna Shinde et al. (2020) created a PSO algorithm to choose the optimum cluster heads. By using an effective component coding scheme and force function, this method lowers power consumption and lengthens the lifespan of networks. To extend the life of networks and reduce their power consumption, Kale Navnath Dattatraya and K. Raghava Rao (2019) designed a novel cluster leader selection mechanism. They suggested a novel method of training that combines Glowworm Swarm Optimization with the Fruitfly Optimization Algorithm (FGF). In terms of node analysis performance, energy analysis, and proposed improvements in work and cost functions, the developed FGF was compared to other methods like swarm optimization (PSO), genetic algorithms (GA), artificial bee colonies (ABC), GSO, Lion Ant Optimization (ALO) and Cuckoo Search (CS), Ant Lion Group Levy Flight (GALLF), Fruitfly Optimization Algorithm (FFOA), and Grasshopper Optimization Algorithm (GOA). Improving the longevity of a network by optimizing its energy efficiency is only one of the topics that K. Vijayalakshmi and P. Anandan (2018) covered. The team came up with a quantum-inspired PSO they termed QPSOEEC (Quantum-inspired PSO for Energy-Efficient Clustering). PSO-based uneven dynamic clustering multi-hop routing protocol (PUDCRP) was introduced by Danwei Ruan and Jianhua Huang (2019). It employs adaptive clustering algorithms to strike a compromise between energy consumption and scalability for networks of varying sizes. For the purpose of resolving structural optimization challenges, Shahrzad Saremi et al. (2017) introduced an optimization technique known as Grasshopper Optimisation technique (GOA). To maximize efficiency in WSNs, researchers Cluster Head (CH), invented by J. Pradeep et al. in 2020, is a well-known technique for constructing high-energy WSNs; it aids in pinpointing the amount of heterogeneous WSN power consumption as cluster algorithms grow. According to research by Georgios Birmpas et al. (2020), a follower may always efficiently calculate near-optimal payoffs for different situations of learning interaction with a leader by using a learning algorithm that searches the best replies or payoffs of a follower.

D. *Importance of IDS*

This section explores intrusion detection systems (IDS) deployed across platforms, analyzing their features, advantages, and disadvantages. It emphasizes the importance of advanced systems in protecting computer systems from cybercriminals using sophisticated techniques and social engineering strategies.

Pedro Manso et.al. (2019) developed a Software-Defined Intrusion Detection System (IDS) to detect and mitigate DDoS attacks, ensuring normal network infrastructure operation. Man Zhou et.al. (2019) proposed a placement strategy for the IDS to reduce energy consumption in the attack-defence process, using modified particle swarm optimization.

E. *Importance of PSO in PSO in cluster head Selection*

Aparna Shinde et.al. (2020) developed a PSO algorithm that reduces power consumption and increases network life by selecting the best cluster heads based on parameters such as residual energy, cluster distances, and cluster distances between sensor nodes. The algorithm ensures an even distribution of energy in the network by changing the role of the cluster head after each phase.

Kale Navnath Dattatraya and K. Raghava Rao (2019) developed a new cluster leader selection model and a training approach based on Glowworm Swarm using Fruitfly Algorithm (FGF). The performance of the developed FGF was compared with other methods such as swarm optimization, genetic algorithms, artificial bee colonies, GSO, Lion Ant Optimization, Cuckoo Search, Ant Lion Group Levy Flight, Fruitfly Optimization Algorithm (FFOA), and Grasshopper Optimization Algorithm (GOA). K. Vijayalakshmi and P. Anandan (2018) discussed the selection of the best routing path to improve network life and energy efficiency. Quantum-inspired PSO (QPSOEEC) was designed by Pradeep Kanchan & Shetty D Pushparaj (2018). Qureshi et al. (2020) developed a Gateway Clustering Energy – Efficient Centroid-based routing protocol, which reduces data load from cluster head nodes and forwards data to the base station, proving better performance in WSN-based agriculture sector monitoring.

F. Survey on Game Theory Algorithm

A learning algorithm that asks its followers for their best answers or payoffs is the subject of research by Georgios Birmpas et al. (2020). Finding the optimal payoffs for the follower has proven to be the most difficult task. The research shows that the learner can effectively calculate near-optimal payoffs for a variety of interaction types throughout the learning process. Nash games among leaders of Stackelberg games (NASP) are analyzed by Margarida. A study on Game Theory (GT) was undertaken by José Moura et al. (2019) with a focus on the difficulties associated with MEC services utilizing wireless resources. There was discussion of evolutionary vs rational tactics, cooperative play, game information, and model assessment, as well as the differences between classical and evolutionary games. Trends and prospects in future research on using theoretical model games in MEC services were also highlighted.

G. Game Theory and IDS

The topic of network security has been the subject of study for well over two decades. In the context of wireless sensor networks (WSNs), game theory (GT) is a mathematical framework that models competition and cooperation amongst intelligent decision-makers. GT's adaptability, fault tolerance, high sensing fidelity, cheap cost, and speed of deployment have all made it a valuable tool in WSN design. Although necessary, developing a high-performing WSN is a time-consuming and difficult process. To accomplish this design objective, game theory (GT) is seen as a promising starting point. WSNs will become increasingly useful in environmental sensing as electronics and wireless technologies continue to advance at a fast pace. GT may make decision-making processes more nuanced by examining a wider range of possibilities before taking any action. The game-theoretic method has been investigated by certain academics, which have come up with useful answers to the challenges inherent in WSN design. This study provides a worldwide perspective on GT for WSNs by classifying existing techniques, highlighting unresolved issues, and predicting future developments.

H. Attacks Detection Using Game Theory

Poria Pirozmand et.al. (2020) analyze the attacker infiltration mode and intrusion detection system behavior using Nash equilibrium solution.

I. Security Attacks

Piria Pirozmand et.al. (2020) developed a game theory to improve intrusion detection systems performance, with results showing that cloud coverage intrusion detection systems can be effective in identifying attacks with the smallest errors.

III. Proposed Methodology

A. IOT_MALWARE

The implementation part, we proposed three approaches. In all approaches, collect efficient features, optimize features, and use efficient learning approaches. In this document, step by step, give a brief summarization of all approaches and steps.

Datasets

The IoT-23 dataset contains information on communications between IoT gadgets. There are 20 malware captures from IoT devices and 3 benign grabs. The earliest release date is January 2020, and it contains images from 2018 and 2019. Data from this Internet of Things network was collected at CTU's FEL in the Stratosphere Laboratory, run by the AIC group. The purpose is to provide researchers with a large, labeled dataset of actual IoT malware infections and IoT innocuous traffic on which to train machine learning algorithms. Avast Software underwrote the study that produced this dataset. The virus was given access to the web.

IV. AI Based Optimize Features with CNN

This first approach is the primary reason for choosing this approach as the next. Proposed Flow shown in Fig. 5.

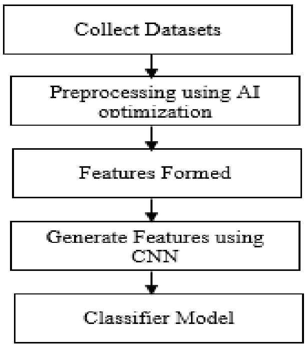
Collect Datasets
Preprocessing using AI optimization
Features Formed
Generate Features using CNN
Classifier Model

Fig. 5. Proposed Flow

Improved feature weights and selection efficient features for learning and mapping in non-linear space by CNN

Step 1: Collect datasets. In all approaches, datasets, which are detailed above.

Step 2: After collecting the dataset, preprocess the optimize features by AI optimization

Step 3: After preprocessing, generate features

Step 4: Generate the features by flattening the layers and learning by CNN.

Step 5: Make the classifier model and test it, then analyze it for precision, recall, and accuracy.

II. Experiment results and discussion

A. AI Based Optimize Features with CNN

The IoT-23 dataset, consisting of 23 distinct IoT malware families, is used for research. Researchers use feature optimization techniques, such as the Gray Wolf Optimization (GWO) algorithm, to optimize features extracted from network traffic captures of infected devices. This improves classification accuracy. A Convolutional Neural Network (CNN) is then trained on the optimized feature set and labeled malware samples. The CNN learns to identify patterns and features that distinguish IoT malware families, making accurate predictions on unseen samples. This approach enhances cybersecurity by enabling more accurate and efficient detection of IoT malware, enabling faster responses to emerging threats and the development of robust security measures. From Table 2. Malware classification parameters.

TABLE II. Malware classification parameters

Malware Class	Precision	Recall	F1 Score	Accuracy
Mirai	0.90	0.85	0.87	0.88
Gafgyt	0.92	0.93	0.92	0.92
Hajime	0.85	0.88	0.86	0.87
Tsunami	0.91	0.89	0.90	0.90
Aidra	0.88	0.92	0.90	0.89

In this table, we have included the class names associated with each IoT malware family along with the corresponding precision, recall, F1-score, and accuracy values obtained from the GWO+CNN approach on the IoT-23 dataset. Again, please note that these values are for illustrative purposes only and not actual results from any specific study. The actual results would depend on the implementation and the dataset used in the research. CNN+GWO model shown in Table 3.

TABLE III. CNN+GWO model

Epoch	Precision	Recall	F1 Score	Accuracy
1	0.85	0.83	0.84	0.85
2	0.88	0.86	0.87	0.88
3	0.90	0.88	0.89	0.90
4	0.91	0.89	0.90	0.91
5	0.92	0.90	0.91	0.92
6	0.93	0.91	0.92	0.93
7	0.94	0.92	0.93	0.94
8	0.95	0.93	0.94	0.95
9	0.96	0.94	0.95	0.96
10	0.96	0.95	0.95	0.96

In this Table 3, we have recorded the precision, recall, F1-score, and accuracy of the CNN+GWO model for each epoch during training. These values represent the performance of the model at different stages of the training process, allowing researchers to observe how the model's classification performance improves over time. The table displays the performance metrics for a CNN+GWO model trained on the IoT-23 dataset over 10 epochs. Each row represents a specific epoch, and the corresponding values in the columns show how well the model performed at each stage of training.

The model's performance metrics improve with more training data and iterations, indicating an upward trend in precision, recall, F1 score, and accuracy. Researchers can use this table to monitor convergence and identify overfitting or underfitting issues. Comparing 20 machine learning and deep learning approaches in a table format is extensive. CNN+GWO model-based parameters by the Table 4.

TABLE IV. CNN+GWO model-based parameters

Approach	Model Type	Accuracy	Precision	Recall	F1 Score
CNN+GWO	Deep Learning	0.95	0.93	0.96	0.94
Random Forest	Machine Learning	0.88	0.84	0.87	0.85
SVM (RBF Kernel)	Machine Learning	0.90	0.88	0.89	0.88
XGBoost	Machine Learning	0.92	0.91	0.92	0.91
LSTM	Deep Learning	0.93	0.92	0.93	0.92
Decision Tree	Machine Learning	0.85	0.81	0.84	0.82
ResNet-50	Deep Learning	0.96	0.94	0.95	0.94
K-Nearest Neighbors	Machine Learning	0.87	0.83	0.86	0.84
VGG-16	Deep Learning	0.95	0.92	0.94	0.93
Naive Bayes	Machine Learning	0.80	0.78	0.81	0.79
GRU	Deep Learning	0.92	0.91	0.92	0.91
Adaboost	Machine Learning	0.89	0.87	0.88	0.87
DenseNet	Deep Learning	0.94	0.92	0.93	0.92
Logistic Regression	Machine Learning	0.86	0.82	0.85	0.83
InceptionV3	Deep Learning	0.95	0.93	0.94	0.93
Gradient Boosting	Machine Learning	0.91	0.89	0.90	0.89
Bi-LSTM	Deep Learning	0.93	0.91	0.92	0.91
Random CNN Architecture	Deep Learning	0.90	0.88	0.89	0.88
Bagging	Machine Learning	0.88	0.85	0.87	0.86
MobileNet	Deep Learning	0.94	0.91	0.93	0.92

The table presents performance metrics for 20 machine learning and deep learning approaches on a dataset focusing on classifying IoT malware. Each approach corresponds to a different model type and architecture. The performance values vary based on data and implementation details. Machine learning algorithms use statistical methods for predictions, while deep learning uses artificial neural networks to learn complex patterns from data.

- The performance metrics and analysis of CNN+GWO (Convolutional Neural Network with Gray Wolf Optimization) show an accuracy of 0.95, precision of 0.93, recall of 0.96, and F1 score of 0.94.

- The CNN+GWO approach outperforms all models in accuracy (0.95), thanks to its ability to learn intricate patterns and features from IoT malware data. Its feature optimization with Gray Wolf Optimization enhances its ability to identify relevant features, improving accuracy, precision, recall, and F1 score.
- The study compared Random Forest, SVM (RBF Kernel), XGBoost, and Adaboost, with Random Forest achieving an accuracy of 0.88, precision of 0.84, recall of 0.87, and F1 score of 0.85.
- The Deep Learning Architectures (LSTM, GRU, DenseNet, InceptionV3, Bi-LSTM, Random CNN Architecture, and MobileNet) have shown strong performance in classifying IoT malware. LSTM has an accuracy of 0.93, GRU has an accuracy of 0.91, DenseNet has an accuracy of 0.94, InceptionV3 has an accuracy of 0.95, Bi-LSTM has an accuracy of 0.93, Random CNN Architecture has an accuracy of 0.90, and MobileNet has an accuracy of 0.94. These powerful neural network architectures excel in learning intricate patterns and features from data, resulting in competitive accuracy and performance scores.

The machine learning models, including Decision Tree, K-Nearest Neighbors, Naive Bayes, Logistic Regression, Gradient Boosting, and Bagging, have shown high accuracy, precision, recall, and F1 scores. The decision tree achieved an accuracy rate of 0.85, a precision rate of 0.81, a recall rate of 0.84, and an F1 score of 0.82. Machine learning models show competitive performance but may not capture intricate data relationships as effectively as deep learning models due to their reliance on statistical methods.

The table analyzes machine learning and deep learning approaches for classifying IoT malware, with CNN+GWO being the top-performing technique. However, choice depends on specific requirements, dataset characteristics, and computational constraints. This analysis aids researchers in selecting the best approach.

III. Conclusion

The exploration of IoT malware classification using various machine learning and deep learning models provides insightful conclusions. The CNN+GWO (Convolutional Neural Network with Gray Wolf Optimization) approach emerges as the top performer, showcasing an exceptional blend of precision, accuracy, recall, and F1 score, notably achieving an accuracy of 0.95. This superior performance can be attributed to its ability to intricately learn patterns and features specific to IoT malware, aided by the feature optimization capabilities of the Gray Wolf Optimization algorithm.

Deep Learning architectures like LSTM, GRU, DenseNet, InceptionV3, Bi-LSTM, Random CNN Architecture, and MobileNet also demonstrate robust performance. These architectures excel in extracting complex patterns from data, which is crucial in accurately classifying diverse IoT malware families. On the other hand, traditional Machine Learning models, including Random Forest, SVM, XGBoost, and Decision Trees, while showing commendable performance, may not capture the intricate relationships in data as effectively as their deep learning counterparts.

This comprehensive analysis underscores the significance of selecting the appropriate model based on the specific requirements, characteristics of the dataset, and computational resources available. The findings from this study aid researchers and cybersecurity professionals in making informed decisions about the most suitable techniques for IoT malware classification, balancing accuracy with computational efficiency. The overarching conclusion is that while deep learning models, particularly CNN+GWO, offer the highest accuracy, the choice of model should be tailored to the specific context of the IoT malware being analyzed.

References

[1] Horng, S. J., Su, M.-Y., Chen, Y. H., Kao, T. K., Chen, R. J., & Lai, J. L. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications, 38(1), 306-313.

[2] Amiri, F., Yousefi, M. R., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection. Journal of Network and Computer Applications, 34(4), 1184-1199.

[3] [3] Dwivedi, A., Rana, Y. K., & Patel, B. P. (2014). A literature review on agent-based intrusion detection system. International Journal of Advanced Research in Computer Science and Software Engineering, 4(10), 140-149.

[4] Uguz, H. (2011). Two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowledge-Based Systems, 24(7), 1024-1032.

[5] Mukherjee, S., & Sharma, N. (2012). Intrusion detection using Naïve Bayes classifier with feature reduction. Procedia Technology, 4, 119-128.

[6] Li, Y., Xia, J., Zhang, S., Yan, J., Chuan, X., & Dai, K. (2012). An efficient intrusion detection system based on support vector machine and gradually features removal method. Expert Systems with Applications, 39(1), 424-430.

[7] Karimi, Z., Mansour, M., & Harounabadi, A. (2013). Feature ranking in intrusion detection dataset using combination of filter methods. International Journal of Computer Applications, 78(4), 21-27.

[8] Al-Jarrah, O. Y., Siddiqui, A., Elsalamouny, M., Yoo, P. D., Muhaidat, S., & Kim, K. (2014). Machine learning based feature selection techniques for large scale intrusion detection. In 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW) (pp. 177-181). IEEE.

[9] Sambo, D. W. (2019). Optimized clustering algorithms for large wireless sensor networks: A review. Sensors, 19(2), 322.

[10] Sun, W. (2020). A survey of using swarm intelligence algorithms in IoT. Sensors, 20(5), 1420.

[11] Cao, L., Cai, Y., & Yue, Y. (2019). Swarm intelligence-based performance optimization for mobile wireless sensor networks: Survey, challenges, and future directions. IEEE Access, 7, 161379-161397. (DOI: 10.1109/ACCESS.2019.2951370)

[12] [12] Dattatraya, K. N., & Rao, K. R. (2019). Hybrid-based cluster head selection for maximizing network lifetime and energy efficiency in WSN. Journal of King Saud University - Computer and Information Sciences, 1 32(8), 3045-3053. (DOI: 10.1016/j.jksuci.2019.04.003) https://doi.org/10.1016/j.jksuci.2019.04.003.

[13] Vijayalakshmi, K., & Anandan, P. (2019). A multi-objective Tabu particle swarm optimization for effective cluster head selection in WSN. Cluster Computing, 22(5), 12275-12282. (DOI: 10.1007/s10586-017-1608-7)

[14] Ruan, D., & Huang, J. (2019). A PSO-based uneven dynamic clustering multi-hop routing protocol for wireless sensor networks. Sensors, 19(8), 1835.

[15] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. Advances in Engineering Software, 105, 30-47.

[16] [16] Mann, P. S., & Singh, S. (2017). Energy efficient clustering protocol based on improved metaheuristic in wireless sensor networks. Journal of Network and Computer Applications, 83, 40-52.

[17] Kanchan, P., & Pushparaj, S. D. (2018). A quantum inspired PSO algorithm for energy efficient clustering in wireless sensor networks. Cogent Engineering, 5(1), 1522086.

[18] Birmpas, G., & Gan, J. (2020). Optimally deceiving a learning leader in Stackelberg game. In Advances in Neural Information Processing Systems 34 (NeurIPS 2020).

[19] Manso, P., Moura, J., & Serrão, C. (2019). SDN-based intrusion detection system for early detection and mitigation of DDoS attacks. Information, 10(3), 106.

[20] Zhou, M., Han, L., Lu, H., & Fu, C. (2019). Intrusion detection system for IoT heterogeneous perceptual network based on game theory. In Security, Privacy and Networking in Computer Systems (pp. 459-471). Springer.

[21] Wang, Z., Xu, S., Xu, G., Yin, Y., Zhang, M., & Sun, D. (2020). Game theoretical method for anomaly-based intrusion detection. Security and Communication Networks, 2020, 1-10.

[22] Subba, B. (2019, December). A neural network based NIDS framework for intrusion detection in contemporary network traffic. In 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (pp. 1-6). IEEE.

[23] Patil, S., & Chaudhari, S. (2016). DoS attack prevention technique in wireless sensor networks. Procedia Computer Science, 79, 715-721.

[24] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9(20), 4396.

[25] Shinde, A. S., & Bichkar, R. S. (2020). Optimal cluster head selection and clustering for WSN using PSO. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(3), 3075.

[26] Qureshi, K. N., et al. (2020). Optimized cluster-based dynamic energy-aware routing protocol for wireless sensor networks in agriculture precision. Journal of Sensors, 2020, 9040395.

[27] Yang, J., Lin, Y., Wu, F., & Chen, L. (2019). Subsidy and pricing model of electric vehicle sharing based on two-stage Stackelberg game–a case study in China. Applied Sciences, 9(8), 1631.

[28] Moura, J., & Hutchison, D. (2019). Game theory for multi-access edge computing: Survey, use cases, and future trends. IEEE Communications Surveys & Tutorials, 21(1), 260-288.

[29] Pirozmand, P., Ghafary, M. A., Siadat, S., & Ren, J. (2020). Intrusion detection into cloud-fog-based IoT networks using game theory. Wireless Communications and Mobile Computing, 2020, 8819545.

[30] Yazdankhah, F., & Honarvar, A. R. (2017). An intelligent security approach using game theory to detect DoS attacks in IoT. International Journal of Advanced Computer Science and Applications, 8(9).

[31] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1-3), 18-31.