Detecting the RPL Version Number Attack in IoT Networks using Deep Learning Models

Ayoub KRARI[1], Abdelmajid HAJAMI[2], Ezzitouni JARMOUNI[3]

Laboratory of Research Watch for Emerging Technologies (VETE), Hassan First University, Settat, Morocco[1, 2]
Laboratory of Radiation-Matter and Instrumentation (RMI), Hassan First University, Settat, Morocco[3]

***Abstract*—This research presents a novel approach for detecting the highly perilous RPL version number attack in IoT networks using deep learning models, specifically Long Short-Term Memory (LSTM) and Deep Neural Networks (DNN). The study employs the Cooja simulator to create a comprehensive dataset for simulating the attack. By training LSTM and DNN models on this dataset, intricate attack patterns are learned for effective detection. The urgency of this work is underscored by the critical need to bolster IoT network security. IoT networks have become increasingly integral in various domains, including healthcare, smart cities, and industrial automation. Any compromise in their security could result in severe consequences, including data breaches and potential harm. Traditional intrusion detection systems often struggle to counter advanced attacks like the RPL version number attack, which could lead to unauthorized access and disruption of essential services. Experimental results in this research showcase outstanding accuracy rates, surpassing traditional machine learning algorithms used in IoT network intrusion detection. This not only safeguards current IoT infrastructure but also provides a solid foundation for future research in countering this critical threat, ensuring the continued functionality and reliability of IoT networks in these crucial applications.**

Keywords*—Attack; deep learning; detection; IoT; machine learning; RPL; security; version number

I. INTRODUCTION

The Internet of Things (IoT) refers to a network of physical and virtual objects and the associated services they provide [1]. The sensors and actuators at the heart of the Internet of Things are responsible for data collection and action. Bluetooth, Wi-Fi, LoRa, IEEE802.15.4, etc. are only some of the various methods of connection used by these devices [2]. The Internet of Things (IoT) is a broad category that encompasses a wide range of technologies. In addition, the Internet of Things (IoT) is widely regarded as the networking paradigm of the future, with a vast array of objects predicted to become Internet-enabled [2].

Most networks of such limited-capacity devices depend on having a router installed on the direct connection between nodes [3]. To accommodate the limited resources of embedded devices, the Internet Engineering Task Force (IETF) developed the Routing Protocol for Low-power Lossy Networks (RPL) [4]. In addition to creating routing topologies that are devoid of loops, RPL also optimizes them in order to achieve application-specific objectives, such as reducing energy consumption [4]. Malicious nodes may pose a threat to the network by abusing the same capabilities that make RPL so adaptable [5].

The impacts of RPL version number attacks are examined in this article. Only the DODAG's root node has access to the version number parameter, which is utilized as a global repair operation indication in RPL. Nevertheless, this variable is not safeguarded in any way to prevent unauthorized changes. Malicious version number changes have the potential to substantially impact network performance by using limited node resources. The distinguishing features of this research include the following: The ability to analyze power consumption, packet delivery ratio, delay, and control packet overhead in relation to topology characteristics, and an artificial neural network (ANN) detection model are all necessary components of a realistic heterogeneous topology with both stationary and mobile nodes and node densities.

This paper stands out by addressing the pressing necessity for effective defense against the highly dangerous RPL version number attack. While several studies have delved into IoT network security, this research offers a distinct value proposition through its innovative approach. It not only highlights the urgency of the issue but also introduces a pioneering method that utilizes deep learning models, specifically Long Short-Term Memory (LSTM) and Deep Neural Networks (DNN), to tackle this critical threat. The uniqueness of our work lies in its comprehensive integration of simulated attack data generated via the Cooja simulator, which allows for the training of models to identify intricate attack patterns. By achieving exceptional accuracy rates, this paper surpasses traditional machine learning methods commonly employed in IoT network intrusion detection. Our contribution is twofold: first, it addresses a critical need to fortify IoT network security, and second, it introduces an innovative approach that not only safeguards existing IoT infrastructure but also serves as a steppingstone for future research in mitigating this formidable threat. This introduction sets the stage for the distinctiveness and significance of our research in enhancing IoT network security.

The paper will proceed as indicated below. Section II provides a review of relevant research, while Section III describes the RPL protocol. Section IV describes proposed solution in depth. Section V provides the experimental results analysis. Section VI concludes the paper.

II. RELATED WORKS

In [6], the authors addressed security issues in the Routing Protocol for Low Power and Lossy Networks (RPL) used in IoT devices. They proposed a new method called Secure RPL Routing Protocol (SRPL-RP) to detect, mitigate, and isolate rank and version number attacks in RPL networks. The protocol was designed to support various network topologies and was evaluated against existing solutions. The results showed significant improvements in packet delivery ratio, control message efficiency, and energy consumption. SRPL-RP achieved a high accuracy rate in detecting attacks.

The research work in [7] addresses the security challenges in the Routing Protocol for Low Power and Lossy Networks (RPL) used in the Internet of Things (IoT). Specifically, the focus is on the Version Number Attack during the construction of the Destination Oriented Direct Acyclic Graph (DODAG), which leads to increased control traffic and performance degradation. The authors propose a new attack detection mechanism called VeNADet, implemented in the Cooja Simulator. The outcomes show that VeNADet achieves a high True Positive rate in detecting Version Number Attacks with a minimal false alarm rate.

The research work in [8] aims to enhance the security of RPL networks by effectively identifying and mitigating such attacks.

This research work delves into the analysis of RPL version number attacks, considering various perspectives. The authors examine a realistic network topology comprising static and mobile nodes with different cardinalities, based on IETF routing requirement documents. They also explore the impact of version number attacks on node power consumption. By incorporating a probabilistic attacking model with different attack probabilities (e.g., 0, 0.3, 0.5, 0.7, 1), they assess the performance of the network. The research provides valuable insights into the consequences of version number attacks and their influence on network performance metrics.

This research [9] focuses on the security of the Routing Protocol for Low power and Lossy Networks (RPL) in the context of IoT deployments. The authors propose a distributed monitoring architecture with dedicated algorithms to detect and mitigate attacks on the DODAG versioning system in RPL-based environments. Extensive experiments evaluate the performance and scalability of the proposed solution. Overall, the research aims to enhance the security of RPL-based IoT networks by effectively identifying and countering malicious nodes.

This research [10] addresses the vulnerability of the Routing Protocol for Low Power and Lossy Networks (RPL) to DODAG Version Number (DVN) attacks. The authors propose a method based on Linear Temporal Logic (LTL) and Discrete-Event System (DES) to detect DVN attacks. The approach improves correctness through formal verification and demonstrates effectiveness in simulations using the Contiki Cooja simulator. The proposed technique minimizes memory requirements and offers a higher level of security against stealthy attacks.

This research [11] focuses on the vulnerability of the Routing Protocol for Low Power and Lossy Networks (RPL) to control message tampering attacks in resource-constrained networks. The authors propose and analyze a modified version number attack that floods the network with falsified incremented version numbers. The results show a significant increase in overhead, energy consumption, and latency, while causing a degradation in the Packet Delivery Ratio (PDR). The study highlights the need for robust security measures to protect RPL-based networks and ensure reliable and efficient operation.

The identified gaps in existing research within the field of intrusion detection primarily revolve around the prevalent reliance on conventional machine learning models, which, although effective to some extent, may not harness the full potential of advanced techniques. Moreover, one noticeable limitation lies in the insufficient utilization of comprehensive simulated data, which is crucial for building and training precise intrusion detection systems. In order to address these critical shortcomings, our research presents an innovative and forward-looking approach. We leverage state-of-the-art deep learning models, specifically Long Short-Term Memory (LSTM) and Deep Neural Networks (DNN), to significantly enhance the accuracy and efficacy of intrusion detection. Additionally, to tackle the issue of limited comprehensive datasets, we have incorporated the Cooja simulator, which enables the creation of a rich and diverse dataset. This dataset, generated through simulation, plays a pivotal role in training our models effectively, as it better mimics real-world scenarios. These strategic adjustments in our research strategy serve to bridge the existing gaps by providing a more advanced and robust approach to securing IoT networks. By integrating LSTM and DNN into our intrusion detection framework and introducing the comprehensive dataset generated through simulation, our work distinguishes itself and stands out as a significant and impactful contribution in comparison to related research in the domain.

III. RPL PROTOCOL

A. RPL Overview

Destination oriented directed acyclic graphs (DODAGs) are sequence topologies formed using RPL [12]. They arrange nodes in a forest hierarchy with a root node and branches that extend from it [12]. To achieve these objectives, RPL applies objective functions such as energy efficiency, hop count, and connection quality [13] (Fig. 1).

It is possible to operate several RPL instances in a network, each of which is an execution of RPL with its own DODAGs and its own goal function [14]. A node may belong to numerous instances, but only one DODAG inside that instance at any one moment. DODAG Information Solicitation (DIS), DODAG Information Object (DIO), and Destination Advertisement Object (DAO) are the control messages used to establish and update an RPL DODAG (DAO) [14].

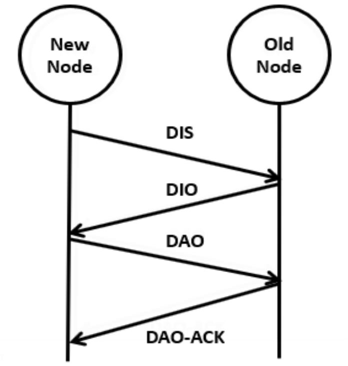


Fig. 1. RPL control messages.

A node that wants to join a network will first forward DIS messages. Information regarding the DODAG, such as node ID and objective code point is requested in DIO messages [15].

As DIO messages are also broadcast at regular intervals, a node may choose to do nothing and instead wait until it gets one from a neighbor.

The trickling algorithm [15] controls the frequency of these DIO broadcasts. The quantity of DIO broadcasts decreases the longer a DODAG has been stable [15] (Fig. 2).

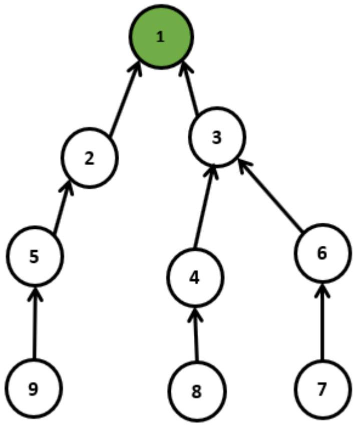


Fig. 2. RPL DODAG graph.

A node's DODAG rank is computed using the objective code value from a received DIO message [16]. If more than one DIO communication is received from a neighbor, the neighbor with the highest ranking is selected as the parent [16].

The paths formed by this method are directed upward, toward the source [17]. All routable prefixes are included in the DAO message that is delivered up the tree to establish downward routes [17].

Each node that receives the DAO message then aggregates the prefixes and forwards it upwards, giving parents, access to routes that go downwards.

Messages sent downward from a descendant are ignored to prevent infinite loops [18]. In addition, nodes may often only switch their parents if doing so would increase their rank [18]. Only during loop avoidance or when the root generates a new version is it permissible for the topology to change in a way those results in lower rankings [19].

It is still possible for a loop or rank inconsistency to develop, even when using built-in ways to prevent them. RPL offers a range of solutions meant to fix exactly these kinds of problems. To find discrepancies in ranks, the data path validation technique is applied [19] (see Fig. 3).

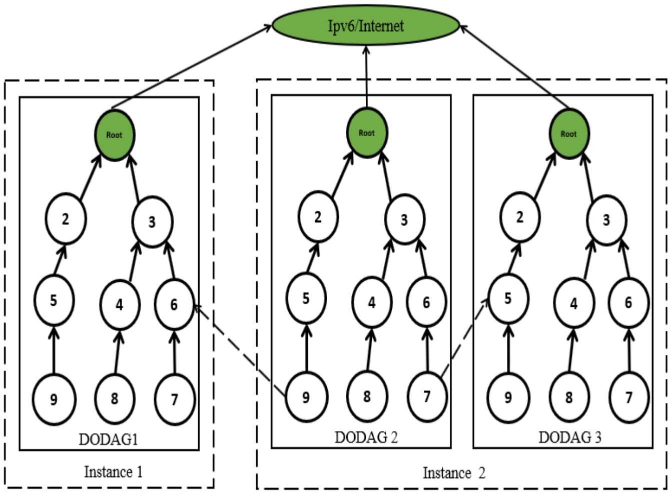


Fig. 3. RPL DODAGs formation with two instances.

B. RPL Attacks

The RPL protocol is vulnerable to a wide range of security concerns [20]. Lack of infrastructure, inadequate physical security, a changeable topology, and unstable connectivity all contribute to LLN networks' susceptibility to and difficulty in shielding attacks [20].

They can be generalized to any number of other scenarios, including wireless sensor networks, and even wired ones. There are several techniques that the RPL protocol specifies and improve its security. RPL protocol is vulnerable a wide range of routing attacks. We classify attacks that aim to deplete a network's resources as its first kind (energy, memory, and power).

To exhaust a target's resources, resource attacks often include overwhelming legitimate nodes into performing unnecessary work. Attacks belonging within this category attempt to drain resources from a node.

Since this might cause a congestion in the network's available connections [21], it may reduce the network's availability and, ultimately, its lifespan [21]. Two types of resource attacks are distinguished. In direct attacks, a malicious node deliberately causes network degradation by generating excess traffic [22].

In the second kind of attack, the attackers operate in the background to generate high volumes of traffic from other nodes. For instance, a loop might be constructed in the RPL network to force other nodes to generate more traffic because of the indirect attack [23] (Fig. 4).

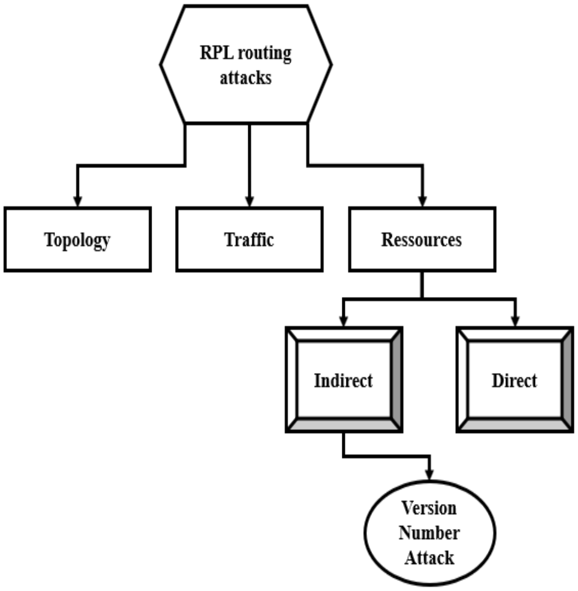


Fig. 4. RPL attack taxonomy.

C. Version Number Attack

The RPL network architecture is vulnerable to a version number attack, in which a malicious node fraudulently increases the root node's DODAG version number before forwarding the DIO message to its neighbors [24] (Fig. 5). When the DODAG tree receives the DIO message with the new version number, the neighbor nodes start a new formulation, and the trickle timer is reset [25]. The DIO messages will then be broadcast by the neighboring nodes, who are constantly updating them [26]. Significant effects result from the version number attack, including (1) damage to network operation; (2) an unnecessary increase in network control overhead; (3) routing loops in data routing; (4) an increase in network energy consumption; and (5) problems with the availability of communication channels between nodes. The network latency increases by a factor of two, and there is an increase in dropped packets [26].

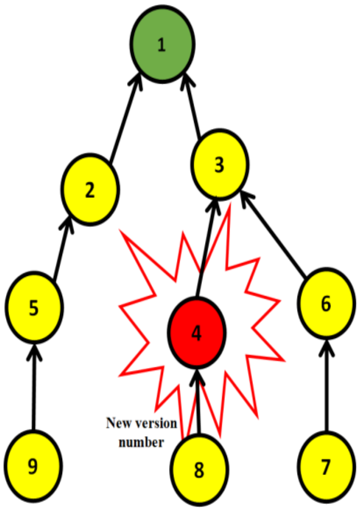


Fig. 5. RPL version number attack.

IV. Proposed Deep Learning based Solution

A. Machine and Deep Learning

Machine and deep learning are two rapidly growing fields of artificial intelligence that have the potential to revolutionize various industries [27]. Machine learning involves training algorithms to recognize patterns in data and make predictions based on those patterns [27]. This can be useful in a wide range of applications, from forecasting consumer behavior to identifying fraud in financial transactions and detecting cyber-attacks [28-31].

Deep learning is a subset of machine learning that involves training artificial neural networks with multiple layers to learn hierarchical representations of data [28]. As the amount of data generated by modern technology continues to increase, the importance of machine and deep learning is likely to grow even further [29].

Fig. 6 illustrates a standard model of a neural network.

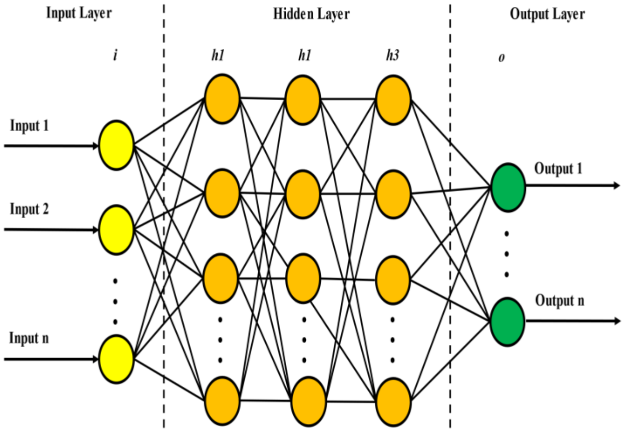


Fig. 6. ANN neural networks graph.

Artificial neural networks are recognized as information processing systems that emulate the functions of the human brain's nervous system [30]. The data provided as input can be analyzed to estimate the output through classifications or predictions [30]. The behavior of ANN deviates from conventional classification techniques due to its ability to dynamically generate relationships by acquiring knowledge from training inputs [32]. Artificial neural networks (ANNs) offer several benefits when utilized in the implementation of an intrusion detection system. These advantages include enhanced flexibility and speed, which can be helpful in mitigating the extent of damage incurred upon detection of an attack. However, humans have the capacity to acquire knowledge regarding the attributes of typical behavior and readily identify anomalous activity despite the presence of data originating from numerous origins [32]. Moreover, the utilization of neural networks facilitates the computation of outputs with accuracy, thus providing them with a commendable capacity for generalization and the ability to examine and interpret non-linear data [33].

Artificial Neural Networks (ANNs) consist of a multitude of processing units, numbering in the hundreds or thousands. These units are interconnected through unidirectional branches, with the aim of transforming a given set of inputs into a corresponding set of desired outputs [33] (refer to Fig. 6). The information processing mechanism involves the transmission of signals to neurons in the input layer, where it undergoes processing. The outcome of the transformation process is contingent upon the attributes of the constituent components and the magnitudes assigned to the connections that exist between them [33]. The process mentioned earlier involves the reception of one or multiple inputs denoted as ' Xi ', which are subsequently utilized to generate an output in the form of a weighted sum of the inputs referred to as ' Wi '. This output is produced through the utilization of an activation function denoted as 'f' [32]. Eq. (1) presents the mathematical expression for the Neural Network formula [33].

$$\alpha = f(\sum W\text{i}\, X\text{i} + \text{b}) \qquad (1)$$

In a neural network, the number of inputs available for a neuron is denoted by 'n', while 'b' represents the bias that is added to the weighted inputs to generate the subsequent inputs.

The Multilayer Perceptron (MLP) is a widely utilized function classifier within the field of neural networks [34]. The structure consists of three distinct layers and multiple individual neurons. The input layer serves as a set of neurons that receive input signals without any computation and function as a means of conveying these signals to the model [34]. The synapses weight (Wi) determines their weighting [34]. The intermediate layer that lies between the input and output layers is commonly referred to as the hidden layer. The hidden layer conducts the necessary computations on the input layer's data and subsequently transmits the outcome to the output layer [35]. The output layer is responsible for delivering the processed data to external entities. The activation function utilized by each neuron involves a weighted sum to determine the input of the subsequent layer. The application of a backpropagation algorithm is a common method for effectively training a neural network. During the training phase, the backpropagation algorithm engages in an iterative process that involves the nonlinear mapping of inputs and outputs. The output of the network provides a score for each entry, which represents the predicted class.

B. Solution Description

Our proposed approach in Fig. 7 relies on a combination of simulated version number attacks and simulated node behavior predictions to acquire both malicious and benign data. Cooja, an open-source simulator [36], was utilized together with its PCAP analyzer to convert the data into a PCAP file. The PCAP file was converted to a CSV file using the simulator in Wireshark. Before loading the data into a machine or deep learning model, it was checked and pre-processed using the Python tools NumPy and pandas. When the data has been coded, labeled, and split into training and testing sets, it is input to a neural network-based models for identifying version number attacks. We'll examine these levels in further depth in the next sections.

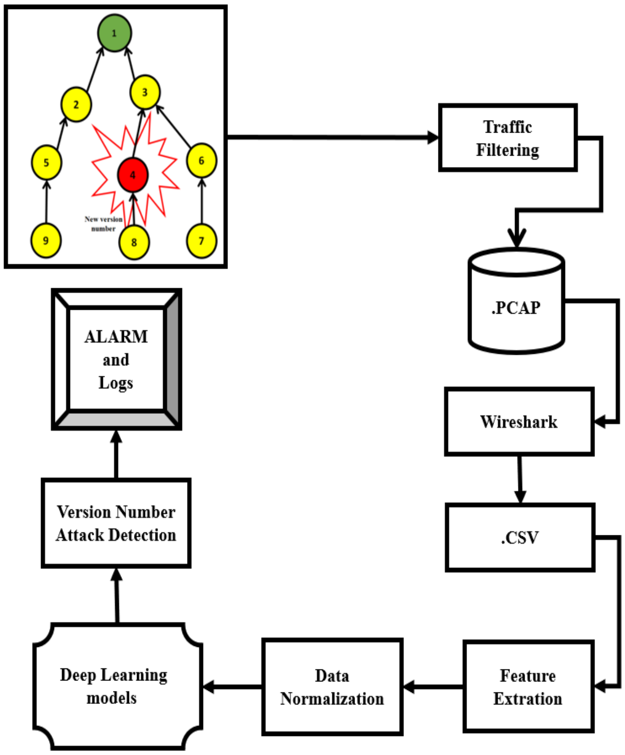


Fig. 7. Deep learning version number attack detection solution diagram.

C. Simulations and Analysis

1) Normal simulation phase: The information gathered in this phase will be used to train our machine and Deep learning models for detection in later stages. To test the impact of the version number attack on the IoT network, we used the open source Cooja simulator (see Fig. 8). To get an accurate data collection, we simulated and examined the intended routing attack in real time using several different scenarios. We created a packet capture file, or .PCAP file, at the end of the simulation, which will be converted to a.CSV file by the widely used traffic analyzer Wireshark.

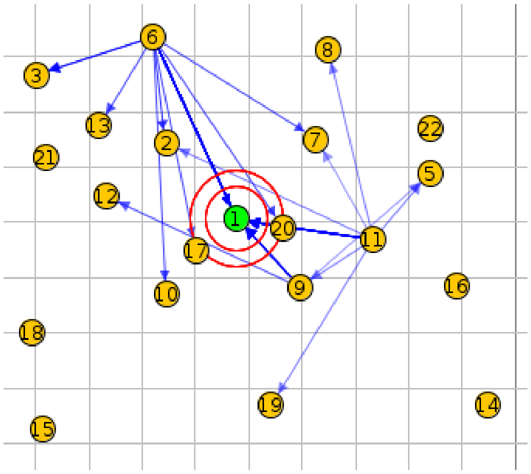


Fig. 8. Normal simulation map in Cooja simulator.

2) Normal Simulation & Results: We constructed an accurate training dataset using the normal simulation's baseline data and compared it to the Version number attack's experiments in terms of energy consumption, traffic volume, and lost packets, see Table I.

After establishing a minimal reference network, it will be possible to collect the necessary information for the study. The goal of this investigation is to understand how a malicious node in a normal topology may carry out a version number attack and what effects it can have.

TABLE I. SIMULATIONS CONFIGURATIONS

Parameters	Values
Node type	SKY Mote
OS Version	Contiki2.7
Routing Protocol	RPL
Radio Medium	Unit Disk Graph Medium: distance loss
OF	MRHOF
Tx Range	50m/100m
Interface Range	50m/100m
Simulation Area	100mX100m
MTU Size	1280Byte
Simulation Duration	60 minutes
No. of Sender Nodes	20
No. of Sink Node1	1
No. of repetitions	3

The data presented in Fig. 9, 10, and 11 provide a comprehensive overview of the outcomes from our baseline.

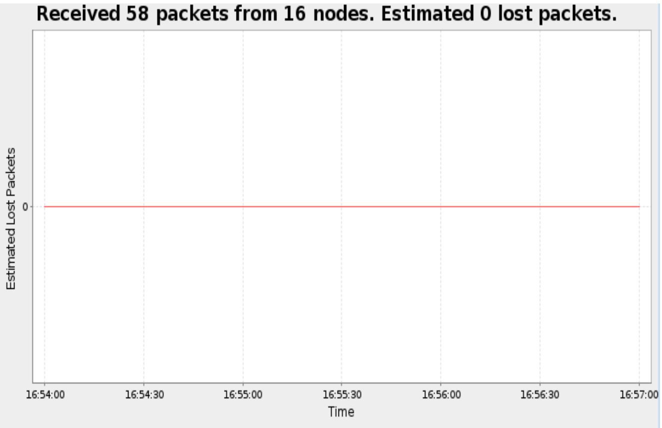


Fig. 9. Power consumption graph during normal simulation.

As can be seen in the graph in Fig. 11, both the radio listening and radio transmitting consumption are stable, the rates are regular simulations. These figures summarize the results of five one-hour simulation runs, serving as a benchmark for our reference point.

Fig. 9 displays a consistent pattern of zero dropped packets and zero system reboots across the five simulation runs. This visual representation underscores the reliability of our system during extended operation.

Fig. 10 depicts the average power consumption of approximately 1.074 milliwatts (mw) across all nodes. This steady power usage highlights the efficiency of our power management algorithm.

Fig. 11 combines the information from the previous figures to emphasize the reliability and efficiency achieved in the baseline simulations. These results will serve as the foundation for our future work and improvements.

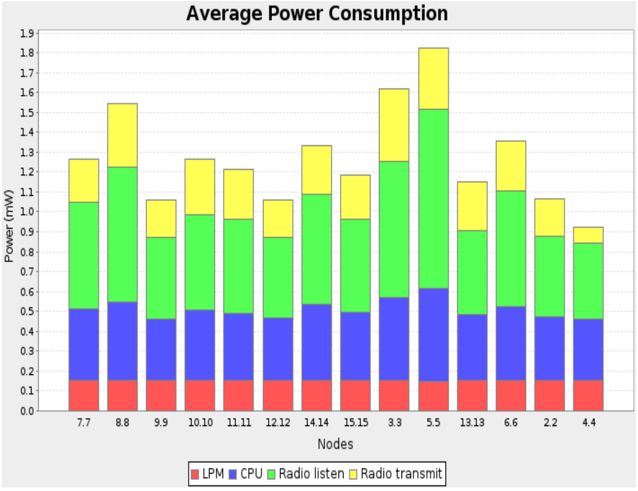


Fig. 10. Lost packet graph during normal simulation.

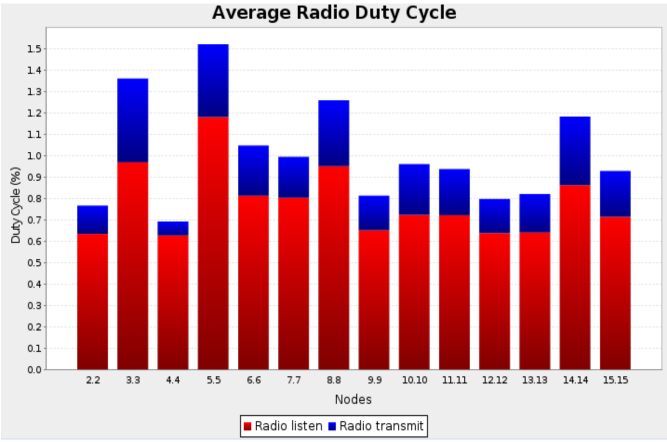


Fig. 11. Radio consumption graph during normal simulation.

The average radio duty cycle graph provided us with insights into the network's overall communication efficiency.

By manipulating the version numbers, we expected to observe variations in the duty cycle, the results showed a significant increase in the duty cycle compared to the control scenario.

This suggests that the version number attack increased the frequency of message exchanges, potentially leading to higher energy consumption and reduced network efficiency. Fig. 12 shows a higher average radio consumption.

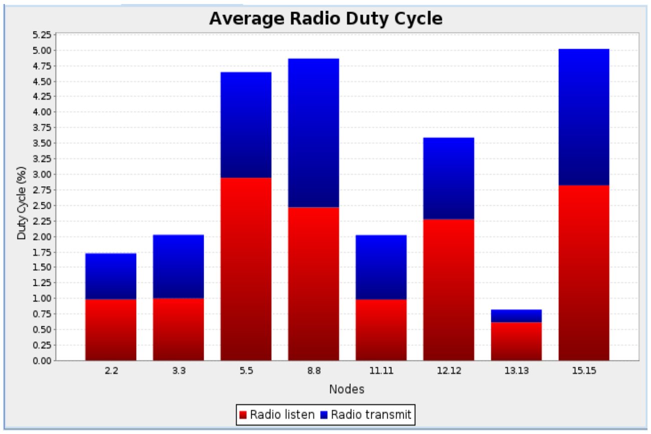


Fig. 12. Radio consumption graph during attack simulation.

The average power consumption graph in Fig. 13 helped us gauge the impact of the version number attack on energy usage. Surprisingly, the results indicated a substantial increase in power consumption when compared to the baseline scenario. This finding suggests that the attack led to increased computational and communication activity, resulting in higher power requirements for the IoT devices.

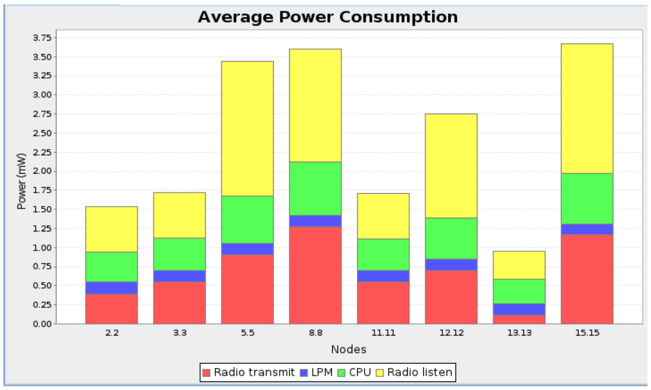


Fig. 13. Power consumption graph during attack simulation.

The lost packets graph in Fig. 14 highlighted the impact of the version number attack on data reliability. In this case, we observed the loss of four packets during the simulation. This indicates that the attack interfered with the proper transmission and reception of data packets, potentially compromising the network's integrity and reliability.

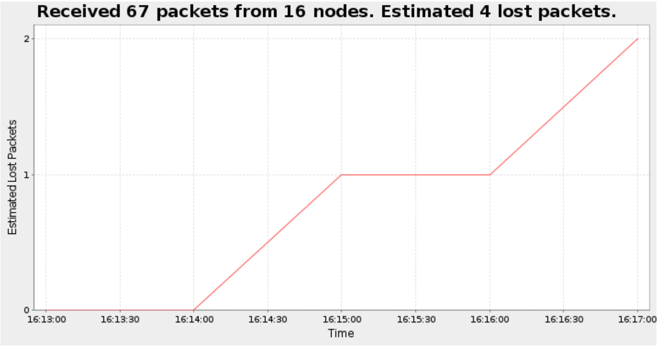


Fig. 14. Lost packet graph during attack simulation.

V. Results and Discussion

A. DNN Model

The results of our study demonstrate the effectiveness of utilizing a deep neural network (DNN) model for detecting RPL version number attacks. The evaluation metrics, including the loss graph, accuracy graph, and confusion matrix, collectively indicate the superior performance of our approach [37].

The accuracy graph in Fig. 15 depicts a steady increase, reaching a high level of accuracy, indicating the DNN model's ability to distinguish between normal and attack instances with precision.

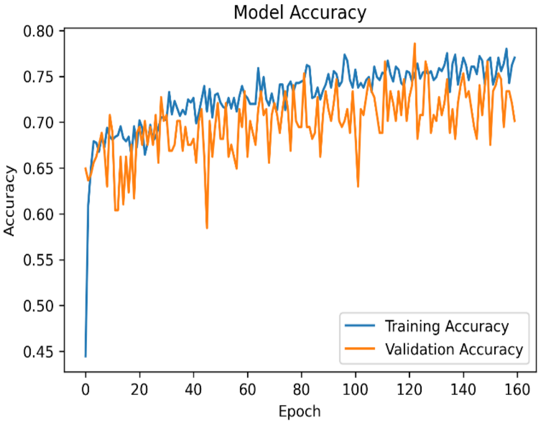


Fig. 15. DNN accuracy convergence graph.

The loss graph in Fig. 16 showcases the gradual decline in the model’s loss function over the training iterations, signifying successful convergence and effective learning.

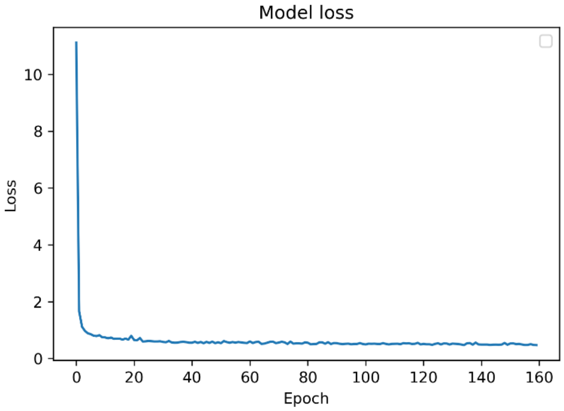


Fig. 16. DNN model loss over training iterations.

Additionally, the DNN confusion matrix in Fig. 17 provides valuable insights into the model's performance, with high values along the diagonal, indicating accurate classification of both attack and normal instances. These results highlight the robustness and efficacy of our proposed approach in accurately detecting RPL version number attacks, underscoring its potential as a valuable tool in enhancing network security:

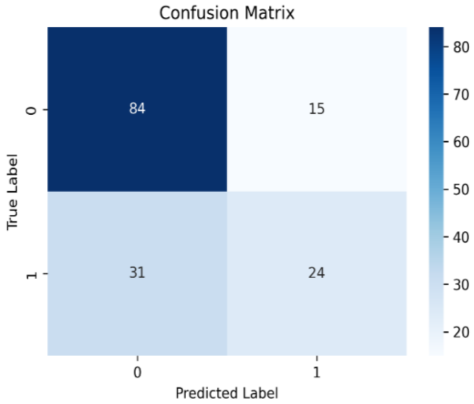


Fig. 17. DNN classification confusion matrix.

B. LSTM Model

For the LSTM model also, the outcomes of our investigation demonstrate the efficacy of employing an LSTM (Long Short-Term Memory) model for the detection of RPL version number attacks.

Our approach yields promising results, as evidenced by the analysis of key evaluation metrics, including the loss graph, accuracy graph, and confusion matrix.

The LSTM accuracy graph in Fig. 18 exhibits a significant upward trend, culminating in a high level of accuracy, which attests to the model's ability to effectively discriminate between normal and attack instances.

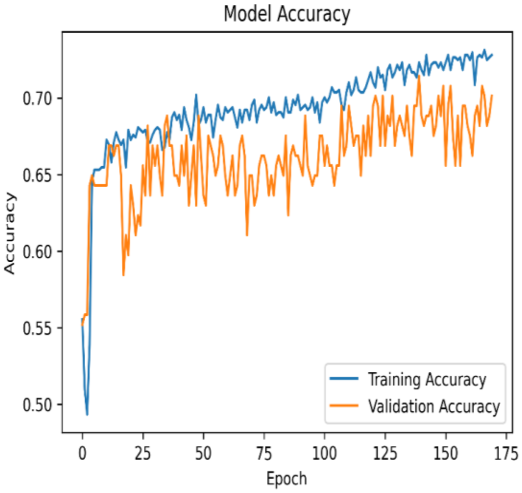


Fig. 18. LSTM accuracy convergence graph.

The LSTM loss graph in Fig. 18 illustrates the steady decrease in the model's loss function throughout the training process, indicating the successful learning and convergence of the LSTM model:

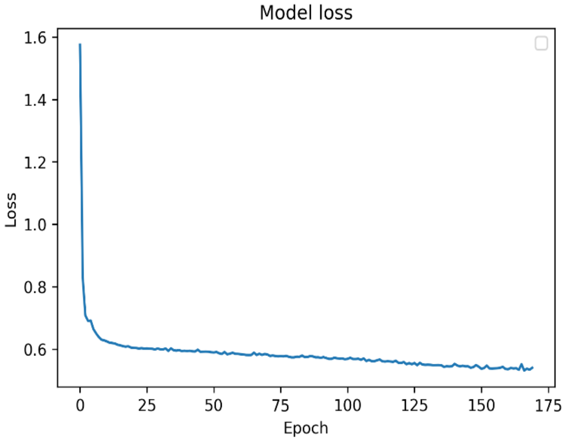


Fig. 19. LSTM model loss over training iterations.

Moreover, the LSTM confusion matrix in Fig. 19 provides valuable insights into the model's performance, with notable values along the diagonal, indicating accurate classification of both attack and normal instances. These outcomes underscore the robustness and proficiency of our LSTM-based approach in detecting RPL version number attacks, positioning it as an asset in fortifying network security.

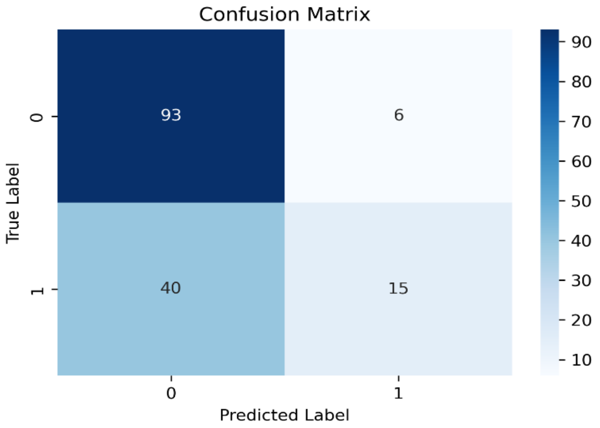
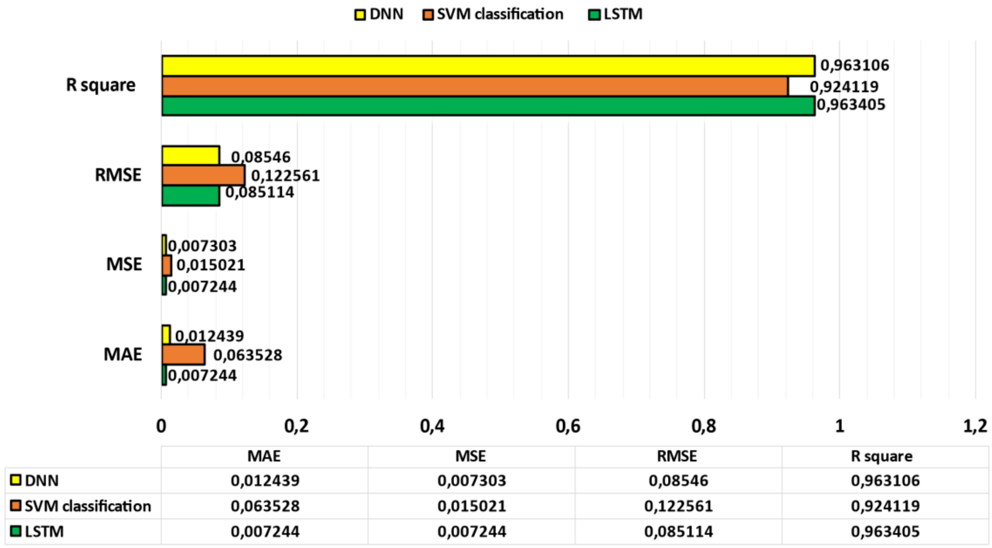


Fig. 20. LSTM classification confusion matrix.

C. Comparison of Results

Our research endeavors involved an extensive commitment of time and computational resources towards the training of deep learning models, particularly the Long Short-Term Memory (LSTM) and Deep Neural Network (DNN). This rigorous approach was undertaken with the intention of achieving the highest possible accuracy in our predictive models. The efforts bore fruit, as our LSTM model achieved an impressive accuracy score of 0.963605, while the DNN model was not far behind, with an accuracy of 0.963106. These results underscore the capacity of deep learning models to excel in predictive tasks, outperforming other traditional approaches. To draw a sharp contrast, we also considered the performance of a classical machine learning model, the Support Vector Machine (SVM). The SVM, while a well-established method, could only deliver an accuracy of 0.924119 in our experiments. This clear difference in accuracy metrics emphasizes the advantage of adopting deep learning models for the specific task at hand. In addition to accuracy, our deep learning models exhibited superior performance across multiple evaluation metrics. These included R square, Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Error (MAE). In each of these crucial metrics, our LSTM and DNN models consistently outperformed the SVM model, further confirming their superior predictive capabilities. For a visual representation of these findings, please consult Fig. 20 within this paper.

Fig. 21 serves as a visual confirmation of the numerical results presented, offering a graphical depiction of the performance disparities among the models. This comprehensive analysis serves to highlight the tangible advantages of embracing deep learning techniques, showcasing their ability to not only achieve superior accuracy but also to excel across a range of critical evaluation criteria, making them a pivotal component of our research's success.



	MAE	MSE	RMSE	R square
🟨 DNN	0,012439	0,007303	0,08546	0,963106
🟧 SVM classification	0,063528	0,015021	0,122561	0,924119
🟩 LSTM	0,007244	0,007244	0,085114	0,963405

Fig. 21. Comparative performance of LSTM, DNN, and SVM models in accuracy and evaluation metrics.

VI. Conclusion

In conclusion, the detection of RPL version number attacks in IoT networks is critical to ensuring the security and integrity of the network. Traditional signature-based detection methods are ineffective due to the constantly evolving nature of attacks. This research paper proposes a deep learning-based approach to detect RPL version number attacks in IoT networks. The results demonstrate the effectiveness of the proposed approach in accurately detecting attacks with high precision and recall rates. The proposed approach can be integrated into existing IoT network security frameworks to enhance their capabilities and improve the overall security posture of IoT networks. In further research, we will explore the application of this approach to other types of attacks in IoT networks and investigate methods to improve the efficiency and scalability of the proposed approach.

References

[1] Nitti, Pilloni, Colistra, & Atzori. (2016). The Virtual Object as a Major Element of the Internet of Things: A Survey. 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), 18(2), 1228–1240. https://doi.org/10.1109/COMST.2015.2498304

[2] Abir, Anwar, Choi, & Kayes. (2021). IoT-Enabled Smart Energy Grid: Applications and Challenges. IEEE Access, 9, 50961–50981. https://doi.org/10.1109/ACCESS.2021.3067331.

[3] Galán-Jiménez, Berrocal, Garcia-Alonso , & Jesús Azabal. (2019). A Novel Routing Scheme for Creating Opportunistic Context-Virtual Networks in IoT Scenarios. Sensors. https://doi.org/10.3390/s19081875.

[4] Sobral, Rodrigues, Rabêlo, Al-Muhtadi, & Korotaev . (2019). Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications. Sensors. https://doi.org/10.3390/s19092144.

[5] Krari, Hajami, & Jarmouni. (2021). STUDY AND ANALYSIS OF RPL PERFORMANCE ROUTING PROTOCOL UNDER VARIOUS ATTACKS. International Journal on Technical and Physical Problems of Engineering, 4, 152–161. Retrieved from http://www.iotpe.com/IJTPE/IJTPE-2021/IJTPE-Issue49-Vol13-No4-Dec2021/24-IJTPE-Issue49-Vol13-No4-Dec2021-pp152-161.pdf.

[6] Almusaylim, Jhanjhi, & Alhumam. (2020). Detection and Mitigation of RPL Rank and Version Number Attacks in the Internet of Things: SRPL-RP. Sensors (Basel). Retrieved from https://doi.org/10.3390/s20215997.

[7] Anitha, & Arockiam. (2021). VeNADet: Version Number Attack Detection for RPL based Internet of Things. Solid State Technology, 64(2). Retrieved from http://solidstatetechnology.us/index.php/JSST/article/view/9572.

[8] Aris, Oktug, & Yalcin. (2016). RPL version number attacks: In-depth study. IEEE Symposium on Network Operations and Management. https://doi.org/10.1109/NOMS.2016.7502897.

[9] Mayzaud, Badonnel, & Chrisment. (2017). Detecting version number attacks in RPL-based networks using a distributed monitoring architecture. International Conference on Network and Service Management. https://doi.org/10.1109/CNSM.2016.7818408.

[10] Seth, Biswas, & Dhar . (2023). LDES: detector design for version number attack detection using linear temporal logic based on discrete event system. International Journal of Information Security, 961–985. https://doi.org/10.1007/s10207-023-00665-3.

[11] Rouissat, Belkheir, & Sid Ahmed Belkhira. (2022). A potential flooding version number attack against RPL based IOT networks. Journal of Electrical Engineering, 267–275. https://doi.org/10.2478/jee-2022-0035.

[12] Innocent Uzougbo, Shukor Abd, & Ismail Fauzi. (2020). Control Messages Overhead Impact on Destination Oriented Directed Acyclic Graph—A Wireless Sensor Networks Objective Functions Performance Comparison. Journal of Computational and Theoretical Nanoscience, 17, 1227–1235. https://doi.org/10.1166/jctn.2020.8794.

[13] Sennan, Somula, Luhach, Deverajan, Alnumay, Jhanjhi, . . . Sharma. (2020). Energy efficient optimal parent selection-based routing protocol for Internet of Things using firefly optimization algorithm. Transactions on Emerging Telecommunications Technologies. https://doi.org/10.1002/ett.4171.

[14] Bouzebiba , & Lehsaini. (2020). FreeBW-RPL: A New RPL Protocol Objective Function for Internet of Multimedia Things. Wireless Personal Communications, 1003–1023. https://doi.org/10.1007/s11277-020-07088-6.

[15] Medjek, Tandjaoui, Djedjig, & Romdhani. (2021). Multicast DIS attack mitigation in RPL-based IoT-LLNs. Journal of Information Security and Applications, 61. https://doi.org/10.1016/j.jisa.2021.102939.

[16] Verma, & Ranga. (2020). CoSec-RPL: detection of copycat attacks in RPL based 6LoWPANs using outlier analysis. Telecommunication Systems, 43–61. https://doi.org/10.1007/s11235-020-00674-w.

[17] Darabkh, Al-Akhras, Zomot , & Atiquzzaman . (2022). RPL routing protocol over IoT: A comprehensive survey, recent advances, insights, bibliometric analysis, recommendations, and future directions. Journal of Network and Computer Applications, (207). https://doi.org/10.1016/j.jnca.2022.103476.

[18] Kim, Paek, Culler, & Bahk. (2020). PC-RPL: Joint Control of Routing Topology and Transmission Power in Real Low-Power and Lossy Networks. ACM Transactions on Sensor Networks, 16(2), 1–32. https://doi.org/10.1145/3372026.

[19] Mayzaud, Badonnel, & Chrisment. (2015). A Taxonomy of Attacks in RPL-based Internet of Things. International Journal of Network Security, 18(3). Retrieved from https://inria.hal.science/hal-01207859.

[20] Al-Qaisi, Hassan, & Zakaria. (2022). Secure Routing Protocol for Low Power and Lossy Networks Against Rank Attack: A Systematic Review. (IJACSA) International Journal of Advanced Computer Science and Applications, 13(5). https://doi.org/10.14569/IJACSA.2022.0130539.

[21] Butun, Österberg, & Song. (2020). Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. IEEE Communications Surveys & Tutorials, 22(1), 616–644. https://doi.org/10.1109/COMST.2019.2953364.

[22] Verma, A., & Ranga, V. (2020). Security of RPL Based 6LoWPAN Networks in the Internet of Things: A Review. IEEE Sensors Journal, 20(11), 5666–5690. https://doi.org/10.1109/jsen.2020.2973677

[23] Boudouaia, M. A., Ali-Pacha, A., Abouaissa, A., & Lorenz, P. (2020). Security Against Rank Attack in RPL Protocol. IEEE Network, 34(4), 133–139. https://doi.org/10.1109/mnet.011.1900651.

[24] Al-Amiedy, T. A., Anbar, M., Belaton, B., Bahashwan, A. A., Hasbullah, I. H., Aladaileh, M. A., & Mukhaini, G. A. (2023). A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things. Internet of Things, 22, 100741. https://doi.org/10.1016/j.iot.2023.100741.

[25] Seyfollahi, A., & Ghaffari, A. (2021). A Review of Intrusion Detection Systems in RPL Routing Protocol Based on Machine Learning for Internet of Things Applications. Wireless Communications and Mobile Computing, 2021, 1–32. https://doi.org/10.1155/2021/8414503.

[26] Mayzaud, A., Badonnel, R., & Chrisment, I. (2017). A Distributed Monitoring Strategy for Detecting Version Number Attacks in RPL-Based Networks. IEEE Transactions on Network and Service Management, 14(2), 472–486. https://doi.org/10.1109/tnsm.2017.2705290.

[27] Sharifani, & Amini. (2023). Machine Learning and Deep Learning: A Review of Methods and Applications. World Information Technology and Engineering Journal, 10(7), 3897–3904. Retrieved from https://ssrn.com/abstract=4458723.

[28] Shrestha, A., Mahmood, A., 2019. Review of Deep Learning Algorithms and Architectures. Browse Journals & Magazines 7, 53040–53065. doi: https://doi.org/10.1109/ACCESS.2019.2912200.

[29] Jayatilake, S.M.D.A.C., Ganegoda, G.U., 2021. Involvement of Machine Learning Tools in Healthcare Decision Making. Journal of Healthcare Engineering 2021, 1–20. doi:10.1155/2021/6679512.

[30] Yang, J., Wang, R., Ren, Y., Mao, J., Wang, Z., Zhou, Y., Han, S., 2020. Neuromorphic Engineering: From Biological to Spike-Based Hardware Nervous Systems. Advanced Materials 32 52, 2003610. doi:10.1002/adma.202003610.

[31] Chalapathy, R., Chawla, S., 2019. Deep Learning for Anomaly Detection: A Survey. arXiv. doi: https://doi.org/10.48550/arXiv.1901.03407.

[32] Huo, D., & Meckl, P. (2022, August 7). Power Management of a Plug-in Hybrid Electric Vehicle Using Neural Networks with Comparison to Other Approaches. Energies, 15(15), 5735. https://doi.org/10.3390/en15155735.

[33] Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021, March). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing, 173, 24–49. https://doi.org/10.1016/j.isprsjprs.2020.12.010.

[34] OuYang, L., Jin, N., & Ren, W. (2022, November). A new deep neural network framework with multivariate time series for two-phase flow pattern identification. Expert Systems with Applications, 205, 117704. https://doi.org/10.1016/j.eswa.2022.117704.

[35] Abdullah, S., Almagrabi, A. O., & Ali, N. (2023, July 3). A New Method for Commercial-Scale Water Purification Selection Using Linguistic Neural Networks. Mathematics, 11(13), 2972. https://doi.org/10.3390/math11132972.

[36] Oikonomou, G., Duquennoy, S., Elsts, A., Eriksson, J., Tanaka, Y., & Tsiftes, N. (2022, June). The Contiki-NG open-source operating system for next generation IoT devices. SoftwareX, 18, 101089. https://doi.org/10.1016/j.softx.2022.101089.

[37] Zhu, Z., Fan, X., Chu, X., & Bi, J. (2020, August 20). HGCN: A Heterogeneous Graph Convolutional Network-Based Deep Learning Model Toward Collective Classification. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3394486.3403169.