ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MACHINE LEARNING TECHNIQUES FOR CYBER SECURITY

SOUMIK SUR[1], MOHAMED EL-DOSUKY[2,3], SHERIF KAMEL[2,4]

[1]Next Tech Lab, SRM University - AP, Andhra Pradesh, India.

[2]Computer Science Department, Arab East Colleges, Saudi Arabia

[3]Computer Science Department, Faculty of Computers and Information, Mansoura University, Egypt

[4]Department of Communications and Computer Engineering, October University for Modern Sciences and Arts, Egypt

E-mail: maldosuky@arabeast.edu.sa

ABSTRACT

Machine learning (ML) is a branch of artificial intelligence (AI) that focuses on developing computer programs that can recognize patterns in historical data, learn from it, and make logical judgements with little to no human input. Protecting digital systems, such as computers, servers, mobile devices, networks, and related data against hostile assaults is known as cyber security. Two key components of combining cyber security with ML are accounting for cyber security where machine learning is used and using machine learning to enable cyber security. This coming together may benefit us in a number of ways, including by enhancing the security of machine learning models, enhancing the effectiveness of cyber security techniques, and supporting the efficient detection of zero day attacks with minimal human interaction. The cyber security landscape has grown more complicated due to the quick development of technology, creating a number of difficulties for protecting sensitive data and important infrastructures. This project's objective is to implement three different systems using machine learning in cyber security. The first system investigates how reinforcement learning may be used to improve cyber security measures. Reinforcement learning algorithms are taught to make the best choices based on their interactions with the environment through trial and error, which can be useful in adjusting to changing cyberthreats. The second approach focuses on malware identification since evasive and polymorphic malware have proven difficult to identify using standard signature-based methods. Several machine learning and deep learning approaches are used in this effort to accurately identify and categorize dangerous software. The third solution uses machine learning and deep learning techniques to address the crucial problem of network intrusion detection. The performance of each system's machine learning models will be evaluated throughout the project using a variety of datasets alongside evaluation measures.

Keywords: *Machine learning, Cyber security, Deep learning, Network, Attack*

1. INTRODUCTION

Machine learning (ML), a sub-field of artificial intelligence, is concerned with creating computer systems that can recognize patterns in historical data, learn from it, and reach logical conclusions with little to no human input. Machine learning techniques have developed into useful tools for enhancing cyber security measures in response to the growing hazards and flaws in our digital world. Due to the rapid advancement of technology and the growing sophistication of cyber attacks, traditional security measures are frequently insufficient to manage the constantly evolving threat landscape [1].

In the modern era of computers, the majority of the gadgets we use are interconnected via the Internet of Things (IoT). Through the Internet, an open and insecure communication medium, these gadgets exchange and send data. These kinds of data such as social security numbers, healthcare information, banking and insurance information are typically sensitive in nature. The malevolent entities, such as the online attackers (hackers), are constantly seeking out the places where they can manipulate things (for instance, they can launch assaults like replay, man-in-the-middle, impersonation, credential guessing, session key computation, malware insertion, and data manipulation) [1].

Cyber security is the practice of implementing security precautions to ensure the confidentiality, integrity, and availability of data. Cyber security is described in the literature on a number of occasions. The ability to ensure and safeguard a nation's information, assets, and vital infrastructure in cyberspace is described by Canongia and Mandarino as "the art of ensuring the existence and continuity of the information society of a nation." Due to the massive amount of data that is collected, processed, and stored by government, military, commercial, financial, and civilian organizations, cyber security is a key research field. Companies need to organize their efforts across their entire information system if they want to be on the defensive side of cyber security. Cyber security consists of a variety of elements, including network security, application security, mobile security, data security, endpoint security, and more [2].

For the early identification and prediction of various assaults, such as spam classification, fraud detection, malware detection, phishing, dark web or deep web sites, and intrusion detection, machine learning techniques are essential in many aspects of cyber security. The lack of qualified employees in certain specialized cyber crime detection fields is a problem that can be solved by using machine learning (ML) approaches. Additionally, aggressive strategies are required to recognize and respond to the new generation's automated and evolving cyber attacks. Machine learning (ML) is one of the potential remedies to act swiftly against such attacks as ML can learn from experiences and promptly respond to newer attempts. To identify and combat cyber attacks, machine learning and deep learning models are currently utilized in practically all aspects of cyber security [3].

A number of difficulties arise when employing machine learning for cyber security, though. The requirement for substantial quantities of labelled data to train machine learning algorithms is one of the major obstacles. When it comes to cyber security, this can be especially problematic because it can be hard to find and access data about emerging dangers. The accuracy of machine learning models may also be limited by the possibility of biassed or inadequate data. The possibility of adversarial attacks is a problem with machine learning's use to cyber security. In order to undermine the security mechanisms that rely on machine learning algorithms, adversarial assaults are made to deceive them into incorrectly classifying data. In the context of cyber security, adversarial attacks can be especially damaging since they may give attackers the opportunity to get around security measures and access private data [4].

Despite these difficulties, machine learning's role in cyber security is anticipated to expand in the future. Models for cyber security that are more precise and efficient are anticipated to be produced as machine learning techniques advance, such as deep learning and reinforcement learning. Additionally, the creation of fresh cyber security datasets and technologies is expected to aid in resolving some of the issues with using machine learning to cyber security. Machine learning techniques could improve cyber security measures and aid in reducing the rising number of cyber threats that both enterprises and individuals must contend with. However, it is impossible to disregard the difficulties in applying machine learning to cyber security. Collaboration between experts in machine learning, data science, and cyber security will be necessary to address these difficulties. By collaborating, it might be possible to create machine learning models that are precise, trustworthy, and useful for increasing cyber security [5].

This paper explores the fundamentals of machine learning, examines various algorithms, highlights the significance of data and feature engineering, and discusses emerging trends like fusing machine learning with blockchain, IoT, and cloud computing. Also discuss the challenges faced by cyber security professionals and examine the fundamentals of machine learning. The importance of explainable AI for transparency in cyber security is also highlighted in the article.

2. LITERATURE REVIEW

Machine learning approaches for cyber security are reviewed in this study. Two types of ML algorithms can be distinguished:

2.1 Supervised learning techniques

For tasks like malware classification and intrusion detection, supervised learning techniques like support vector machines (SVM) and random forests have been used successfully in cyber security. These algorithms provide predictions based on observed patterns by learning from labelled datasets [6]. Support vector machines (SVMs), boosted and bagged decision trees, k-nearest neighbors, Naive Bayes, discriminant analysis, logistic regression, and neural networks are examples of common classification algorithms [7].

2.2 Unsupervised learning techniques

Unsupervised learning techniques, such as clustering algorithms such as k-means and DBSCAN, are useful for spotting irregularities and trends in network data. These methods are especially helpful in identifying new or zero-day threats in situations when labelled data may be limited. A popular unsupervised learning method is clustering. depends on the information that the various learning tasks and the training data give. Additionally, employ Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), two types of deep learning algorithms, have demonstrated outstanding performance in a range of cyber security applications, such as malware identification, phishing detection, and network intrusion detection. These methods excel in discovering deep correlations and automatically learning complex features in huge datasets. This has to do with ML [7,8].

Using these strategies in cyber security poses certain difficulties and problems. Data Quality, Data Quantity, Adversarial Attacks, Interpretability, Explainability, Compatibility Issues, are just a few of the challenges that were discovered. Eavesdropping, traffic analysis, replay attacks, man-in-the-middle attacks, impersonation attacks, denial-of-service (DoS) attacks, malware attacks, scripting attacks, physical theft of smart devices, dictionary attacks, stolen verifier attacks, unauthorized session key computation attacks, attacks on machine learning models, dataset poisoning, model poisoning, privacy breaches, and runtime disruption attacks are just a few of the issues that have been discovered. We will address these problems and obstacles by utilizing machine learning techniques [1].

3. NEWLY PROPOSED SYSTEM

5.1 Reinforcement learning Cyber security

The network represents the environment in which the simulation takes place. Since two different agents coexist in it, it means that we are in the context of a multi-agent environment. There are three types of different nodes. The first node, START: it is the attacker’s starting note, it can be seen as his personal computer. The second node, INTERMEDIATE NODES: they are nodes without sensitive data and must be hacked by the attacker to reach the final node. The end node, DATA: it is the node of the network in which sensitive data are stored. It is the attacker’s target. Each node is characterized by a set of variables that indicate the different attack types the attacker is capable of using, the corresponding defense strategies, and the likelihood that an attack will be detected. An attacker node is specifically specified as n(a_1,a_2,..., a_{10}), where each a represents the attack value of a certain attack type on the node. A node for the defender is specified as n(d_1, d_2,..., d_{10}, det), where each d represents the node's defense against a particular assault type and det represents its detection value. Each node has a detection value, 10 distinct attack kinds, and 10 different protection types. A node's maximum value for each value is 10. Each pair of the attack and defense values, which indicate the assault power and security level of a certain hacking approach, is coupled. The possibility of detecting and apprehending the hacker following a successful attack block is represented by the detection value.

The attacker wants to gain access to the nodes that are between him and the DATA node. Attacking a node m from a node n with a certain attack type, such as ai, is one action the attacker can take. This increases the corresponding attack value in node n by one.

Finding the attacker is the defender's objective. Selecting a node n and a defense type di or the detection value det is one action the defender can take. The matching defense value or detection value in node n is raised by one as a result of performing this.

The network is initialized at the beginning of each, which includes initializing the attack values, defense values, and detection values in each node as given in Table 1.

The two agents taking part in the simulation must figure out the best course of action. Various reinforcement learning techniques have been used to do this: Specifically, for the defender, Deep Q-learning, Monte Carlo learning, and Q-Learning.

Monte Carlo algorithm: Agents learn using Monte Carlo techniques in the first reinforcement learning approach [10]. These methods include averaging the sample returns in order to solve the reinforcement learning challenges. Monte Carlo techniques are only developed for episodic tasks in order to guarantee the availability of well-defined results. When it is possible to describe the state value function in a job using tiny, finite state sets, Monte Carlo techniques are utilized. In actuality, the so-called tabular approaches include Monte Carlo techniques. Using an incremental implementation, Monte Carlo techniques update each state the agent visited with the same reward value:

$$Q_{t+1}(s, a) = Q_t (s, a) + \alpha (R - Q_t(s, a)) \quad (1)$$

where α is the learning rate, a number between 0 and 1 that indicates how much the agent should pick up from a new observation. R is the ultimate payment, either 100 or -100. The estimated reward values, or Q-Values, represent how much the agent anticipates earning after taking a certain action. A particular action is denoted by the letter a, whereas the letter s denotes the current state.

Q-Learning: The same considerations that were established for the Monte Carlo techniques also apply to Q-Learning because it is a tabular method. The primary distinction between the Q-Leaning and Monte Carlo approaches is that the former is an "Online method," which means that updating the Q-Values is possible at any time and does not need waiting until the conclusion of an episode. As a result, sequential settings can also apply Q-Learning. Since the update is performed by first beginning from the most recent state visited and then moving backwards, Backward Q-Learning [11] is utilized instead of Q-Learning as suggested by Elderman et al. [12]. The following state-action pair is updated:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha(R - Q_t(s, a)) \quad (2)$$

Except for the last state-action pair, the learning update is provided by:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha(\gamma \max_b Q_{t+1}(s', b) - Q_t(s, a)) \quad (3)$$

This second formula uses a similar reward system to the original Q-Learning algorithm, but unlike that method, the reward is only delivered at the very end; there are no intermediate rewards. The next state is s'.

Deep Q-learning: Deep Q-learning, which applies Q-learning to deep learning using a deep network called Deep Q-Network (DQN), is the most recent learning algorithm to be used. To fully capitalize on the knowledge the agent has, only the defender is permitted to employ this technique. In actuality, dimensionality limitations prevent the defender's tabular approaches from fully exploiting the environmental state. The neural network trains its weights using stochastic gradient descent back-propagation:

$$Wj = Wj - \eta \frac{\delta J}{\delta wj} \quad (4)$$

where w_J is the loss function, η is the learning rate, and w_j is the jth weight. The incentives are normalized to 1 for success and -1 for failure in order to increase efficiency and stability. The experience replay mechanism is also utilized for training. To hasten convergence, the network is updated by selecting random experiences from a buffer where they were previously stored.

Exploration strategies can be either ε-greedy or SoftMax. In ε-greedy, with a probability of 1–, this approach chooses the optimum course of action; in all other circumstances, it chooses a random course of action from the available options. Here, the value between 0 and 1 represents the degree of investigation. In SoftMax, based on the anticipated reward value of each action, this technique provides each action in the set of potential actions a chance to be chosen.

There are four files divided up for the implementation:

simulation.py: This file's purpose is to manage the simulation. The idea is put into practice in it. as previously mentioned, consisting of a series of time steps that come to an end when one of the termination criteria is met. Both the defense and the attacker select an action for each time step. The two activities ought to be carried out simultaneously, according to theory. Data about the present and subsequent states as well as potential rewards are then saved in various arrays. This is done at the conclusion of the episode to update the Q-values or weights that are utilized by the various reinforcement learning algorithms. Attacker discovery, DATA node hacking, or a tie are the three probable termination scenarios. In reality, there's a slim chance that the attacker won't be found and won't ever succeed in hacking the DATA node.

environment.py: By using a matrix that shows the neighborhood relationships between the nodes (neighbors_matrix) and variables that store information about the number of nodes and the number of attack/defense values per node, it keeps track of all the environmental information, including the attack, defense, and detection values, the indication of whether the attacker has been detected or if the DATA node has been compromised, and the topology of the network. It also reveals the two primary functions that are employed to carry out the movements of both the defense and the attacker.

attacker.py & defender.py: The attacker and defense will be shown together because their implementations are fairly similar. The parameter values of each employed algorithm are stored in a data structure that is unique to each agent. Following some of the recommendations in the reference study [12], the parameters were chosen. In fact, the combination of parameters to get the optimum performance out of each agent versus a rival trained using the Monte Carlo approach and the "greedy" strategy is provided in it. The latter, on the other hand, is optimized for usage against opponents who utilize random choice.

For results, the same experimental settings were used to compare the results to those found in the reference study. In actuality, 10 simulations were run for each algorithm pair. The average of the numerous observations was eventually calculated. The outcomes are displayed in Table 2 below, which compares the different percentages of win and defeat for the two players using different methods.

5.2 Malware detection by machine learning & deep learning

Attacks by malware provide a significant security concern that might impede the widespread use of wireless technologies. Malware attacks are frequent cyberattacks in which the victim's system is compromised by malware, which is often malicious software. Ransomware, spyware, command and control, and other specialized sorts of assaults are all included in malicious software, sometimes known as viruses. Malware deployment has been linked to criminal organizations, state entities, and even well-known corporations; in some situations, it has even been shown to have occurred [Malware Attacks: Definition and Best Practices]. Although the artificial intelligence subfield of "deep learning" has only lately gained popularity, chemical engineers are already aware with its foundational ideas. Artificial neural networks, the algorithms that enable deep learning, are discussed in this article.

The components of implementation can be decomposed of the following components:

a) Dataset: collections of data from the actual world. "malware.csv" is the name of the data collection. Data can be saved in a tabular manner in a CSV file, which stands for comma-separated values. With the exception of the.csv suffix, CSVs resemble standard spreadsheets.
b) Data Gathering and Analysis: Compile a broad and representative dataset of malicious and safe files. Make sure you comprehend the data's structure, format, and any labels or information that may be included.
c) Data cleaning: It involves identifying and handling any duplicates, missing values, or other anomalies in the dataset. This guarantees the data is correct and trustworthy for training.
d) Feature extraction: It is the process of turning raw data into informative feature vectors that depict the properties of files. These features might be static analysis features (such as file size, import table, strings, and entropy) or dynamic analysis features (such as API calls, system calls, and network behavior) for malware identification.
e) Feature selection: Depending on the size of the dataset and the number of features, you may wish to use feature selection techniques to remove unnecessary or redundant features and minimize dimensionality. This can speed up training and enhance model performance.
f) Normalization: By scaling or normalizing the features, you may make sure that they are all on the same scale. This guarantees that the model learns from all features evenly and prevents features with high values from dominating the training process.
g) Split the preprocessed dataset. The validation set is used for hyperparameter tweaking, the training set is used to train the model, and the testing set is used to assess the performance of the finished model. The best group match to a given template is found via Colormap. To detect individual ICs across subjects, typically provide it a list of fitting ICAs and a template IC, such as the blink for the first subject. There are two iterations in this particular technique. The maps that best match the template are found in the first stage. The analysis is then performed using the average of the maps found in the first stage. To discover good parameters, run with plot and display set to True and label set to False. The marking in the IC objects may then be applied by running with labelling enabled. (Running with the labels and plot off has no effect.)

For the results of this subsection, the dataset displays the classification results for 50000 malware and 50000 benign entries. The classification data type is int64. The test has a 99.99% overall accuracy.

5.3 Intrusion detection by machine learning & deep learning

A network security tool called an intrusion detection system (IDS) was first developed to identify vulnerability exploits against specific applications or computers. Network intrusion detection systems (NIDS) are installed at a predetermined location inside the network to monitor all network traffic coming from all connected devices. It carries out an observation of all subnet traffic passing through and compares that traffic to a database of known attacks. The warning can be delivered to the administrator as soon as an attack is detected or unusual behavior is seen. By examining and spotting hostile network behavior, NIDSs can assist in securing our company.

Artificial neural networks are used in deep learning to carry out complex calculations on a massive quantity of data. It is a sort of artificial intelligence that is based on how the human brain is organized and functions. Artificial neurons, often referred to as nodes, make up a neural network, which is organized similarly to the human brain. Three levels of these nodes are arranged next to one another: The hidden layer(s), the output layer, and the input layer. Deep learning algorithms are essential for identifying characteristics and are capable of handling a sizable number of operations for both structured and unstructured data. Deep learning algorithms, however, might be overkill for some jobs that may include complicated challenges since they require access to enormous volumes of data in order to be successful.

The tried types of algorithms are:

a) Convolutional Neural Networks (CNN): CNNs are created for the processing of visual and image data. To automatically recognise and understand spatial patterns in the input data, they employ convolutional layers. In image identification, object detection, and other computer vision problems, CNNs have achieved outstanding results.
b) Recurrent neural networks (RNNs): RNNs are made to handle sequential data, including time series, text, and voice. They can capture temporal relationships in the data because of loops in their architecture that allow information to remain. The vanishing gradient problem, which affects classic RNNs, makes it difficult for them to learn long-range relationships.
c) Long Short-Term Memory (LSTM) Networks are a kind of RNN that deal with the vanishing gradient issue. They are more adapted to collecting long-term dependencies in sequential data because they feature memory cells and gating mechanisms to selectively keep and forget information.
d) MLPs, or multilayer perceptrons, are a great way to begin learning about deep learning technology. MLPs are a kind of feedforward neural networks that contain many layers of activation-function-equipped perceptrons. A completely coupled input layer and an output layer make up MLPs. They may be used to create speech recognition, picture recognition, and machine translation software since they have the same number of input and output layers but may have several hidden layers.

For implementation of this subsection, the code is divided into two pieces. preprocessing.py and intrusion_detection.py are the first and second files, respectively. Convolutional Neural Networks (CNN) are used in this research for intrusion detection. The dataset used for this project is called MachineLearningCVE and it comes from the actual world.

preprocessing.py: Perform data cleaning, feature selection and extraction, data transformation, normalization/standardization, and other operations in this area.

intrusion_detection.py: Take the data from preprocessing.py and use the convolutional neural networks (CNN) approach in this section. Get the network incursion output from this.

For the results of this subsection, it is shown that the capabilities of intrusion detection may be considerably enhanced by using both machine learning and deep learning approaches. While machine learning models are straightforward and effective, deep learning models are superior at identifying complex patterns in network data. The trials' findings provided important new information on the potential of deep learning and machine learning approaches for intrusion detection. Prediction classes include PortScan, Bot, BENIGN, among others. With these, intrusion detection is predicted 100% of the time.

4. CONCLUSION

This research paper has dug into the field of machine learning approaches for cyber security, researching three key systems: reinforcement learning, malware detection by machine learning & deep learning, and network intrusion detection by machine learning & deep learning. We have investigated the possibilities and difficulties of using machine learning to address important cyber security issues through the deployment of these systems. Scalability, interpretability, adversarial assaults, and data privacy issues were just a few of the difficulties we ran across when doing our inquiry while attempting to use machine learning to cyber security. To create effective and durable cyber defense measures, there is a constant need for research and innovation in the sector. Although each system has had success, we understand that there is no one-size-fits-all approach to cyber security. Cyber dangers are always changing, thus it's important to keep up with their adaptability and progress while also investigating new strategies.

As an intelligent and adaptable method of upgrading cyber security systems, reinforcement learning has showed potential. Reinforcement learning models have the potential to enhance threat response tactics and decision-making procedures in dynamic and changing cyber environments by utilizing the capacity to learn from experience and optimize actions based on rewards and penalties.

The malware detection system has proven that deep learning and machine learning models are effective in detecting and classifying malware. Advanced detection procedures are required due to the malware threats' constant growth and growing complexity, and machine learning techniques offer useful tools in this effort. We have established the groundwork for reliable and effective malware detection systems by experimenting with a variety of datasets and feature engineering approaches.

The network intrusion detection system has provided insightful information on protecting organizational networks from unauthorized access and destructive activity. We have created sophisticated intrusion detection systems that can analyze network traffic patterns and spot possible security breaches by utilizing machine learning and deep learning techniques.

This research has nevertheless also highlighted the difficulties and constraints related to machine learning applications in cyber security. Scalability, interpretability, adversarial assaults, and data privacy issues are still significant challenges that must be addressed in more detail and via further study. This research study adds to the amount of information about machine learning methods for cyber security. The examination of emerging patterns and the practical insights gleaned will assist organizations, researchers, and practitioners in the field of cyber security in their continued efforts to safeguard sensitive data and important assets from the always changing world of cyber threats. In order to protect our linked world and guarantee a safer digital future for everyone, it is increasingly important to integrate intelligent machine learning technologies effectively.

5. FUTURE WORK

Future work may entail examining and resolving numerous factors pertaining to each of the deployed systems. Here are some probable directions for the future:

5.1 Reinforcement Learning (RL) for Cyber Security

For improved convergence and performance in cyber security applications, look into more sophisticated policy optimization techniques as Proximal Policy Optimization (PPO), Trust Region Policy Optimization (TRPO), or Soft Actor-Critic (SAC). Examine approaches to incorporate human experience or topic knowledge into RL agents to enhance their capacity for making judgement calls in response to actual cyberthreats. Study and create RL models that can successfully fight off adversarial attacks. Adversarial Reinforcement Learning. To guarantee the stability of RL-based systems, adversarial training and defenses may be especially crucial for cyber security applications.

5.2 Malware Detection by Machine Learning & Deep Learning

Investigate ways to make machine learning and deep learning models easier to grasp and more transparent, allowing security analysts to comprehend the thinking behind the models' judgements, especially in important situations. Given the swiftly evolving nature of cyber threats, increase the effectiveness and speed of malware detection algorithms to operate in real-time or almost real-time circumstances. Explore approaches like transfer learning or unsupervised learning to efficiently identify previously unknown or zero-day malware, as typical signature-based techniques might not be enough in such circumstances.

5.3 Network Intrusion Detection by Machine Learning & Deep Learning

Investigate and contrast several anomaly detection techniques, such as Isolation Forest, Autoencoders, or Variational Autoencoders (VAEs), for accurately identifying network invasions. Explore how federated learning techniques may be used for network intrusion detection in situations where data privacy and centralized data gathering are key problems, such as in IoT networks or geographically dispersed organizations. Create adaptable models that can continually learn from and improve their ability to identify intrusions in order to take into account changing attack patterns and network dynamics.

Look into methods for fusing ideas and expertise from several cyber security fields (such as RL-based defense, malware detection, and intrusion detection) to create collaborative defense mechanisms that can better fend off complex threats. Encourage multidisciplinary research by working with specialists from related domains, such as psychology (user behavior analysis), cryptography, or hardware security, to create more dependable and thorough cyber security solutions. Contribute to the creation of different and standardized datasets to assess the effectiveness of machine learning models for cyber security. Fair comparisons between various strategies would be made easier as a result. Making the implemented systems and models open-source will encourage sharing and cooperation among researchers, allowing others to build upon your work and advancing the discipline.

REFERENCES:

[1] Wazid, Mohammad, Ashok Kumar Das, Vinay Chamola, and Youngho Park. "Uniting cyber security and machine learning: Advantages, challenges and future research." ICT Express 8, no. 3 (2022): 313-321.

[2] Yavanoglu, Ozlem, and Murat Aydos. "A review on cyber security datasets for machine learning algorithms." In 2017 IEEE international conference on big data (big data), pp. 2186-2193. IEEE, 2017.

[3] Shaukat, Kamran, Suhuai Luo, Vijay Varadharajan, Ibrahim A. Hameed, and Min Xu. "A survey on machine learning techniques for cyber security in the last decade." IEEE Access 8 (2020): 222310-222354.

[4] Cinà, Antonio Emanuele, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard A. Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. "Wild patterns reloaded: A survey of machine learning security against training data poisoning." arXiv preprint arXiv:2205.01992 (2022).

[5] Roopak, Monika, Gui Yun Tian, and Jonathon Chambers. "Deep learning models for cyber security in IoT networks." In 2019 IEEE 9th annual computing and communication workshop and conference (CCWC), pp. 0452-0457. IEEE, 2019.

[6] Shaukat, Kamran, Suhuai Luo, Vijay Varadharajan, Ibrahim A. Hameed, and Min Xu. "A survey on machine learning techniques for cyber security in the last decade." IEEE Access 8 (2020): 222310-222354.

[7] Buczak, Anna L., and Erhan Guven. "A survey of data mining and machine learning methods for cyber security intrusion detection." IEEE Communications surveys & tutorials 18, no. 2 (2015): 1153-1176.

[8] Cinà, Antonio Emanuele, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard A. Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. "Wild patterns reloaded: A survey of machine learning security against training data poisoning." arXiv preprint arXiv:2205.01992 (2022).

[9] Roopak, Monika, Gui Yun Tian, and Jonathon Chambers. "Deep learning models for cyber security in IoT networks." In 2019 IEEE 9th annual computing and communication workshop and conference (CCWC), pp. 0452-0457. IEEE, 2019.

[10] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[11] Wang, Yin-Hao, Tzuu-Hseng S. Li, and Chih-Jui Lin. "Backward Q-learning: The combination of Sarsa algorithm and Q-learning." Engineering Applications of Artificial Intelligence 26, no. 9 (2013): 2184-2193.

[12] Elderman, Richard, Leon JJ Pater, Albert S. Thie, Madalina M. Drugan, and Marco A. Wiering. "Adversarial reinforcement learning in a cyber security simulation." In 9th International Conference on Agents and Artificial Intelligence (ICAART 2017), pp. 559-566. SciTePress Digital Library, 2017.

[13] Li, Bai, Changyou Chen, Wenlin Wang, and Lawrence Carin. "Certified adversarial robustness with additive noise." Advances in neural information processing systems 32 (2019).

[14] Aldhyani, Theyazn HH, and Hasan Alkahtani. "Cyber Security for Detecting Distributed Denial of Service Attacks in Agriculture 4.0: Deep Learning Model." Mathematics 11, no. 1 (2023): 233.

[15] Pirbhulal, Sandeep, Wanqing Wu, Khan Muhammad, Irfan Mehmood, Guanglin Li, and Victor Hugo C. de Albuquerque. "Mobility enabled security for optimizing IoT based intelligent applications." IEEE Network 34, no. 2 (2020): 72-77.

[16] Wang, Yu, Jack W. Stokes, and Mady Marinescu. "Neural malware control with deep reinforcement learning." In MILCOM 2019-2019 IEEE military communications conference (MILCOM), pp. 1-8. IEEE, 2019.

[17] Mayer, Christoph P. "Security and privacy challenges in the internet of things." Electronic Communications of the EASST 17 (2009).

[18] Xin, Yang, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng

Gao, Haixia Hou, and Chunhua Wang. "Machine learning and deep learning methods for cybersecurity." Ieee access 6 (2018): 35365-35381.

[19] Xiao, Liang, Geyi Sheng, Sicong Liu, Huaiyu Dai, Mugen Peng, and Jian Song. "Deep reinforcement learning-enabled secure visible light communication against eavesdropping." IEEE transactions on communications 67, no. 10 (2019): 6994-7005.

[20] Bontemps, Loïc, Van Loi Cao, James McDermott, and Nhien-An Le-Khac. "Collective anomaly detection based on long short-term memory recurrent neural networks." In Future Data and Security Engineering: Third International Conference, FDSE 2016, Can Tho City, Vietnam, November 23-25, 2016, Proceedings 3, pp. 141-152. Springer International Publishing, 2016.

[21] Yuan, Xiaoyong, Pan He, Qile Zhu, and Xiaolin Li. "Adversarial examples: Attacks and defenses for deep learning." IEEE transactions on neural networks and learning systems 30, no. 9 (2019): 2805-2824.

[22] Nkiama, Herve, Syed Zainudeen Mohd Said, and Muhammad Saidu. "A subset feature elimination mechanism for intrusion detection system." International Journal of Advanced Computer Science and Applications 7, no. 4 (2016).

[23] Ambusaidi, Mohammed A., Xiangjian He, Priyadarsi Nanda, and Zhiyuan Tan. "Building an intrusion detection system using a filter-based feature selection algorithm." IEEE transactions on computers 65, no. 10 (2016): 2986-2998.

[24] Saxe, Joshua, and Konstantin Berlin. "Deep neural network based malware detection using two dimensional binary program features." In 2015 10th international conference on malicious and unwanted software (MALWARE), pp. 11-20. IEEE, 2015.

[25] Tobiyama, Shun, Yukiko Yamaguchi, Hajime Shimada, Tomonori Ikuse, and Takeshi Yagi. "Malware detection with deep neural network using process behavior." In 2016 IEEE 40th annual computer software and applications conference (COMPSAC), vol. 2, pp. 577-582. IEEE, 2016.

Table 1: Initial Values.

Node	Attack Values	Defense and Detection Values
START	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CPU1	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1]
CPU2	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 1]
DATA	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 1]

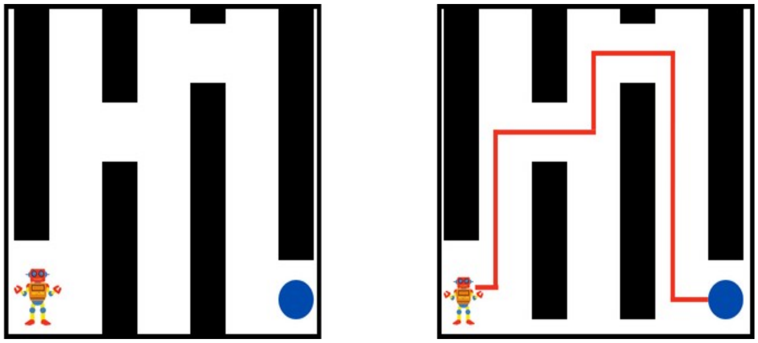

Figure 1: An easy illustration of reinforcement learning is when a robot searches for an object inside a grid area. In reinforcement learning, the robot can be viewed as the agent. The target item is a circle. Obstacles are indicated by the boxes. To locate a path to the circle is the robot's goal

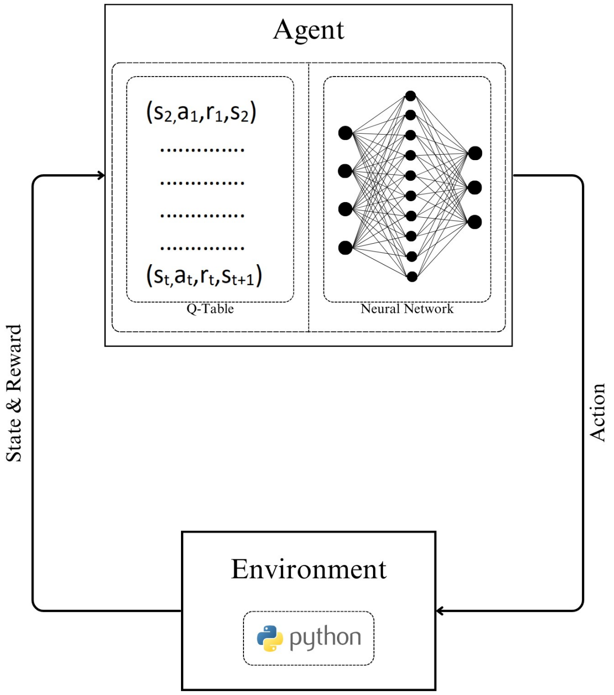

Figure 2: Structure for reinforcement learning

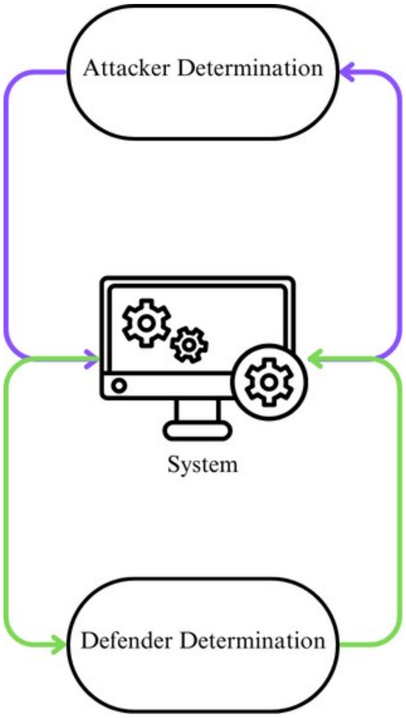

Figure 3: System-based interaction between attackers and defenders

Table 2: Victory/defeat percentages

D ↓ A →	Random	MC ε-greedy	MC Softmax	QL ε-greedy
Random	A: 8.5% D: 91.5%	A: 93.2% D: 6.8%	A: 96.6% D: 3.4%	A: 97.7% D: 2.3%
MC ε-greedy	A: 7.1% D: 92.9%	A: 67.1% D: 32.9%	A: 75.1% D: 24.9%	A: 48.2% D: 51.8%
MC Softmax	A: 5.2% D: 94.8%	A: 70.7% D: 29.3%	A: 79.6% D: 20.4%	A: 65.8% D: 34.2%
QL ε-greedy	A: 8.6% D: 91.4%	A: 62.2% D: 37.8%	A: 67.4% D: 32.6%	A: 42.5% D: 57.5%
DQN	A: 14.4% D: 85.6%	A: 79.8% D: 20.2%	A: 78.9% D: 21.1%	A: 97.4% D: 2.6%

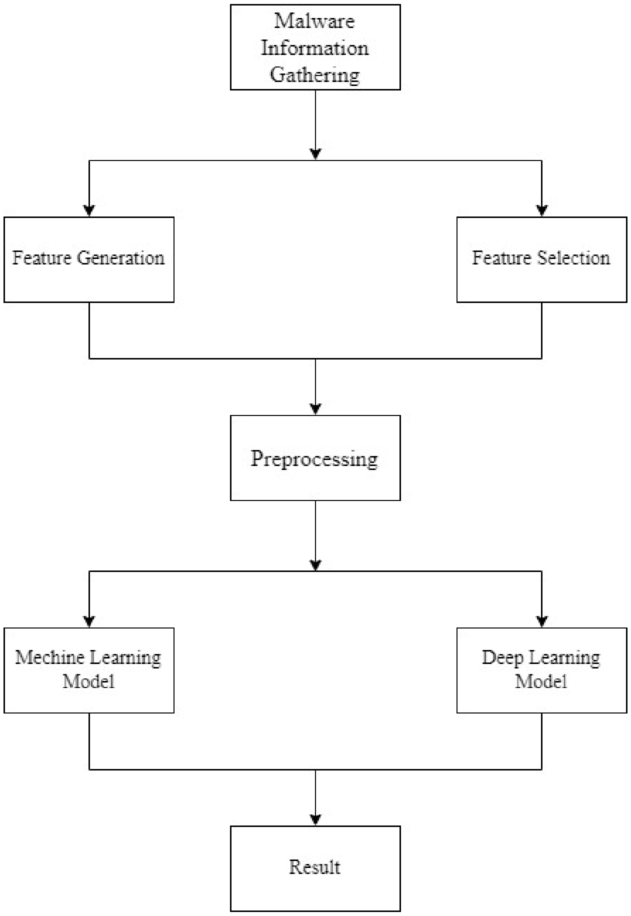

Figure 4: Malware detection architecture

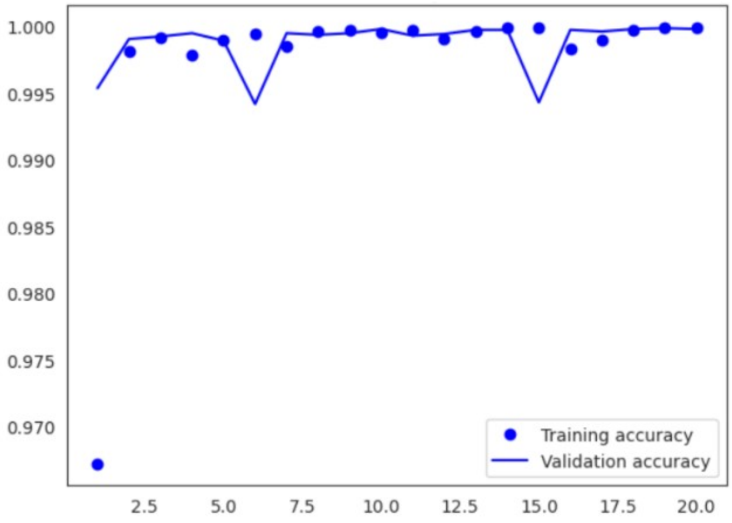

Figure 5: Accuracy

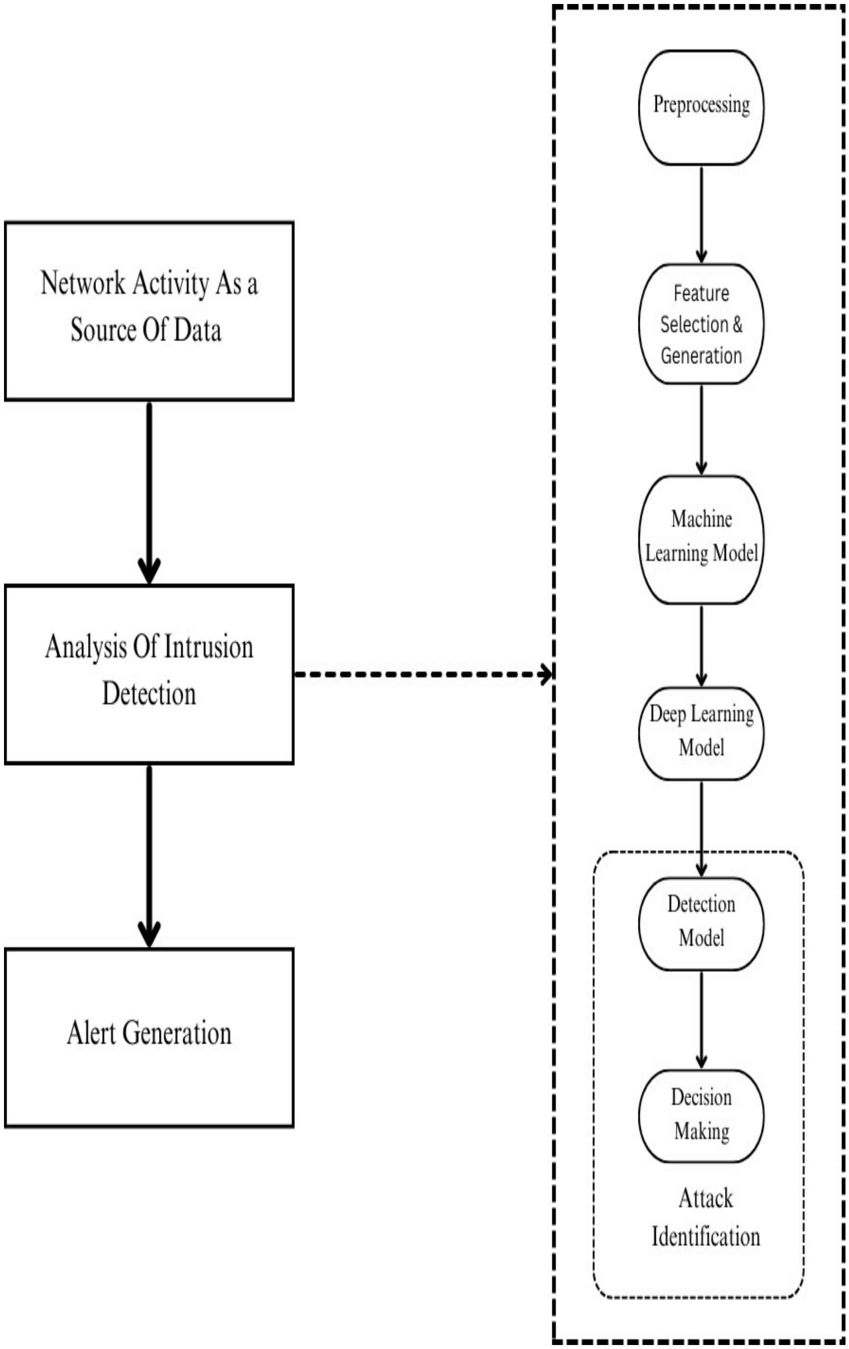

Figure 6: Intrusion detection architecture