Lifelong Machine Learning For Sustainability In The Network Security

Muhammed Emre Komşal
Faculty of Computer and Informatics Engineering
Istanbul Technical University
Maslak/Istanbul
İTÜ Ayazaga Campus, 34469
Email: komsal16@itu.edu.tr

Alper Çetin
Faculty of Computer and Informatics Engineering
Istanbul Technical University
Maslak/Istanbul
İTÜ Ayazaga Campus, 34469
Email: cetina19@itu.edu.tr

Hasan İnanç Güney
Faculty of Computer and Informatics Engineering
Istanbul Technical University
Maslak/Istanbul
İTÜ Ayazaga Campus, 34469
Email: guney19@itu.edu.tr

***Abstract*—The increasing sophistication of network attacks and the rapid evolution of technology have made it challenging for traditional security systems to keep pace with the changing threat landscape. Lifelong machine learning (LML) algorithms have emerged as a promising approach to address this challenge by continuously learning from new data over the entire lifespan of a system. In this research paper, we explore the use of LML algorithms for network attacks to extend the lifespan of network security systems. We review the literature on LML algorithms for network security and discuss different approaches to implementing these algorithms, including supervised, unsupervised, and reinforcement learning. We present a case study on the use of LML algorithms for network intrusion detection and show how they can improve the accuracy and effectiveness of intrusion detection systems. We also discuss the challenges and limitations of using LML algorithms for network security and propose directions for future research. Our findings suggest that LML algorithms can help to extend the lifespan of network security systems by enabling them to adapt and evolve over time.**

1. Introduction

The theft of digital information from public hospitals, banks, the defense industry, educational institutions, or governmental agencies has risen exponentially over the past 20 years, as most industries try to modernize and automate their working cultures in this era of Industry 4.0. While modernizing work environments can bring many benefits, it also creates new vulnerabilities and attack surfaces that can be exploited this time with the use of malware assaults [1]. The word ”malware” is a combination of the words ”malicious” and ”software”. Software is referred to as hostile software when it carries out a routine operation without authorization. A malicious operation is defined as one that takes place without the user's knowledge or that wasn't intended to happen at that specific moment. The two components of malware are the payload (carrier) and the exploits (activity) [2].

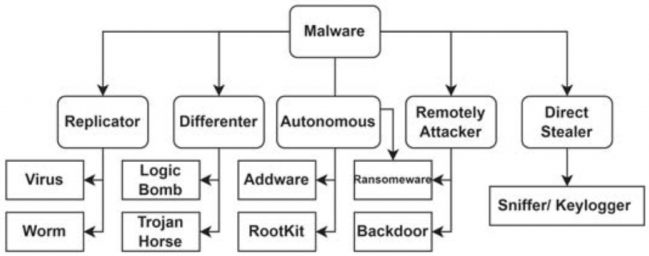

Figure 1. Lifelong Machine Learning [34]

Figure 1 shows many types of malware attacks that are currently available, and whose prevention by utilizing just security mechanisms may be difficult without early discovery. Malware is further classified based on its mode of operation. Furthermore, once malware enters the system and becomes difficult to remove, most attacks are dangerous to the device of the legitimate user [3].

1) Replicator: This virus swiftly replicates itself, consumes all available system memory, and slows down the computer's performance. The main sources of such malware files are.doc files and visiting suspicious websites.

2) Different: It also refers to malware that is programmed to launch at a specific time determined by the hacker or attacker. Such malware enters the system, receives some event triggers, and then begins. Upon triggered, they might either steal data or create a backdoor for hackers to use, such as with logic bombs, horses, etc.

3) Autonomous: Like other types of malware, autonomous malware is not destructive. They do this because their content is available everywhere you look online and because clicking on the adverts takes you to their website. Adware, rootkits, and other types of malware that display adverts on your website are typical examples.

4) Remotely Attacker: In the modern era, this malware attack kind is one of the most effective. Typically, the attacker installs malicious software on the victim's computer

before encrypting all of the system's files or opening a backdoor port to allow remote access to the system. And demand a ransom for the data on behalf of this attacker. Although it is valuable these days, information. For instance, a backdoor, ransomware, etc.

5) Direct Stealer: This malware attack type involves the attacker secretly monitoring a keyboard key and using it to malfunction, such as a sniffer or other software. Individuals who are wealthy or well-known would typically be the targets of such attacks.

Due to its advanced learning and generalization capabilities, machine learning (ML) has been extensively used in a variety of fields, including computer vision, natural language processing, and recommender systems. Some researchers use ML models to categorize malware. These ML-based malware classification techniques are more effective than traditional ones and call for less specialized knowledge [4].

In today's digital age, network security is of utmost importance. The increasing number and sophistication of network attacks have made it essential to have robust and effective security measures in place. According to statistics, almost three networks are attacked every minute, hence network security is the initial component of network development. This is especially important in terms of information security, which can affect both social and economic interests as well as national security. In such a dire situation, understanding how to assess network security is critical [5]. Traditional security systems are no longer sufficient to keep pace with the rapidly evolving threat landscape. Lifelong machine learning (LML) algorithms offer a promising solution by enabling security systems to continuously learn and adapt to new data over the entire lifespan of the system. The process results in the learner gaining knowledge and improving their ability to learn. One of the defining characteristics of human intellect is this capacity for lifelong learning. But the currently dominant ML paradigm learns in isolation: given a training dataset, it just uses the dataset to run an ML algorithm and create a model. It makes no effort to retain what is learned and apply it to new learning. Although this particular ML paradigm, which is mostly based on data-driven optimization, has been very effective, it only works well for certain, narrow tasks in closed contexts and necessitates a large number of training instances [6].

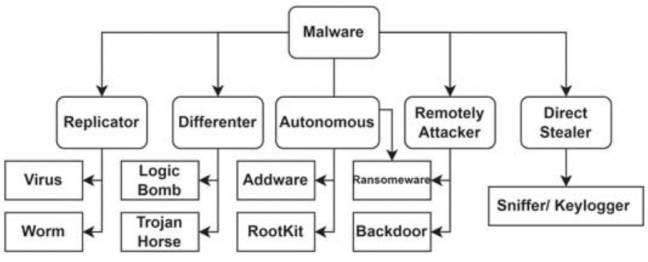

Figure 2. Lifelong Machine Learning [34]

Lifetime self-adaptation is based on the ideas of lifelong machine learning, which provides an architectural solution for a machine learning system's continuous learning. Lifelong machine learning is an addition to a machine learning system that selectively transfers knowledge from previously learned tasks to aid in the learning of new tasks within an existing or new domain. Lifelong machine learning has been combined with a wide range of learning approaches, including supervised, interactive, and unsupervised learning, with success [7].While lifelong machine learning (LML) has several advantages, there are also some potential disadvantages to consider. The amount of computation required by lifelong learning techniques is substantial, especially when compared to conventional machine learning algorithms. Longer training times, greater memory requirements, and greater energy consumption may result from this increasing computational complexity. Additionally, LML models run the risk of underperforming on new data due to overfitting to historical data. The risk of overfitting increases as the model learns more over time. Finally, catastrophic forgetting occurs when the model loses track of information it has already learnt as it picks up new information. Despite the fact that LML approaches work to lessen this risk, catastrophic forgetting is still a possibility and can impair performance on prior tasks [8].

2. Background

A container is a standardized software component that wraps up code and all of its dependencies to ensure that an application will run efficiently and consistently in different computing environments. Containerization provides a clear separation of responsibility, as developers focus on application logic and dependencies, while IT operations teams can focus on deployment and management instead of application details such as specific software versions and configurations. In addition, containers can run virtually anywhere. Greatly easing development and deployment on Linux, Windows, and MacOS operating systems; on virtual machines or on physical servers; on a developer's machine or in data centers on-premises; and of course, in the public cloud. Moreover containers virtualize CPU, memory, storage, and network resources at the operating system level, providing developers with a view of the OS logically isolated from other applications. Containers allow agile development, such as move much more quickly by avoiding concerns about dependencies and environments. It is efficient, containers are lightweight and allow you to use just the computing resources you need. This lets you run your applications efficiently. Also containers are able to run virtually anywhere. As it is seen by positive impact and opportunities given by the containers. It is widely used in today's technologies as well as in the e-health industry.

2.1. Network Security Past Incidents

Network security is a critical aspect of modern technology, as we increasingly rely on connected systems and data

for our daily lives and business operations. Unfortunately, there have been numerous past incidents where network security has been compromised, resulting in significant harm to individuals and organizations. By examining these incidents and their causes, we can better understand the importance of network security and the need for ongoing vigilance and improvement.

2.1.1. Code Red Worm.

One of the earliest and most infamous network security incidents was the Code Red worm in 2001. This worm exploited a vulnerability in Microsoft IIS web servers, allowing it to spread rapidly and cause significant disruption to websites and servers worldwide [9]. The incident highlighted the importance of promptly patching software vulnerabilities and ensuring secure configuration of network systems.

2.1.2. Equifax Data Breach.

In 2017, the Equifax data breach exposed sensitive personal and financial information of over 143 million Americans [10]. The breach was caused by a vulnerability in an open-source software component that Equifax had failed to patch, allowing hackers to gain access to the company's network and steal data. The incident underscored the need for effective vulnerability management and proactive security measures to prevent data breaches.

2.1.3. Target Data Breach.

In a significant 2013 data breach, cybercriminals managed to steal payment card information belonging to Target customers by exploiting a weakness in the network of the company's HVAC contractor [11]. This incident underscored the criticality of performing third-party security assessments and managing supply chain risks. Moreover, it emphasized the necessity for implementing robust access controls and diligently monitoring network activities.

2.1.4. WannaCry Ransomware.

In 2017, the WannaCry ransomware attack affected hundreds of thousands of computers in over 150 countries, causing widespread disruption and financial harm [12]. The attack exploited a vulnerability in Microsoft Windows operating systems that had been patched months earlier, highlighting the risks of failing to apply security updates and the need for effective patch management and contingency planning.

2.1.5. SolarWinds Supply Chain Attack.

In 2020, the SolarWinds supply chain attack compromised the software supply chain of a leading IT management company, resulting in widespread infiltration of customer networks by hackers [13]. The attack demonstrated the vulnerability of supply chain networks and the need for rigorous security controls and monitoring of software supply chains.

2.1.6. Colonial Pipeline Ransomware Attack.

More recently, in 2021, the Colonial Pipeline ransomware attack disrupted fuel supplies for much of the eastern United States, highlighting the potential for cyberattacks to have significant real-world consequences [14]. The attack was caused by a vulnerability in the company's remote access system, underscoring the importance of secure remote access controls and monitoring.

2.1.7. Twitter Bitcoin Scam.

In July 2020, a high-profile Twitter security breach occurred, during which attackers gained access to 130 verified accounts, including those of prominent individuals and companies [31]. The attackers used these accounts to post a Bitcoin scam, tricking users into sending them cryptocurrency. The incident raised concerns about the security of social media platforms and the potential for disinformation and manipulation through compromised accounts.

2.1.8. Microsoft Exchange Server Exploits.

In early 2021, several vulnerabilities in Microsoft Exchange Server, a widely used email and calendar server software, were discovered and exploited by hackers [32]. The vulnerabilities allowed attackers to gain unauthorized access to email accounts, exfiltrate data, and install malware on affected servers. The incident highlighted the importance of promptly patching software vulnerabilities, as well as the need for organizations to monitor and secure their email servers and infrastructure.

2.1.9. Kaseya Ransomware Attack.

In July 2021, a massive ransomware attack targeted the software company Kaseya and its customers, affecting thousands of businesses and organizations worldwide [33]. The attackers exploited a vulnerability in Kaseya's Virtual System Administrator (VSA) software to distribute ransomware to managed service providers and their clients. The incident demonstrated the potential for cyberattacks to have a widespread impact through software supply chains and emphasized the need for robust security measures and continuous monitoring of network systems.

2.2. Past Studies

2.2.1. Denial-of-Service Attacks.

One of the most common types of attacks against computer networks is the denial-of-service (DoS) attack, which aims to disrupt or disable network services by overwhelming them with traffic. In a study conducted by Lu and colleagues [15], the authors analyzed various types of DoS attacks and

proposed a novel approach to detecting and mitigating them using machine learning algorithms. They demonstrated that their approach achieved higher accuracy and lower false positive rates compared to existing methods. Similarly, a study by Kim and colleagues [16] investigated DoS attacks on Internet of Things (IoT) devices and proposed a lightweight defense mechanism that can be integrated into low-power devices with limited resources.

2.2.2. Phishing Attacks.

Phishing is another common attack technique that targets users' personal information and credentials by tricking them into clicking on malicious links or downloading malware. In a study by Khan and colleagues [17], the authors analyzed various characteristics of phishing emails and proposed a machine learning-based classifier that can accurately identify phishing emails. They found that features such as the sender's address, subject line, and body content can be used to distinguish phishing emails from legitimate ones. In another study, Abdelhamid and colleagues [18] presented a machine learning-based anti-phishing framework that utilizes multiple features, including visual similarity, URL reputation, and user behavior, to effectively detect and block phishing websites.

2.2.3. Malware Analysis.

Malware is a type of malicious software that is designed to infect and harm computer systems. In a study by Sharma and colleagues [27], the authors proposed a deep learning-based approach for malware detection that utilizes both static and dynamic analysis techniques. They demonstrated that their approach achieved higher accuracy and lower false positive rates compared to traditional signature-based detection methods. Similarly, a study by Gupta and colleagues [28] investigated the use of machine learning algorithms for malware classification and proposed a framework that can accurately classify malware samples based on their behavior.

2.2.4. Wireless Security.

Wireless networks are vulnerable to various types of attacks due to their open nature and lack of physical boundaries. In a study by Shafique and colleagues [29], the authors provided a comprehensive survey of wireless security issues and challenges, including eavesdropping, jamming, and rogue access points. They also discussed various security mechanisms such as encryption, authentication, and intrusion detection/prevention systems. In another study, Li and colleagues [30] proposed a software-defined networking (SDN)-based approach to enhancing wireless security by integrating multiple security functions into a single platform.

2.2.5. Blockchain Security.

Blockchain is a distributed ledger technology that provides a secure and transparent way of recording transactions. However, blockchain networks are not immune to security threats, such as 51% attacks, smart contract vulnerabilities, and consensus failures. In a study by Ronen and colleagues [21], the authors proposed a framework for assessing the security of blockchain networks and identified several key factors that affect their security, such as network size, consensus mechanism, and smart contract design. They also provided recommendations for improving the security of blockchain networks, including regular security audits, bug bounty programs, and community involvement. Similarly, a study by Ali and colleagues [22] investigated the use of blockchain technology for securing Internet of Things (IoT) networks and proposed a blockchain-based architecture that provides secure and decentralized communication between IoT devices.

2.2.6. Cloud Security.

Cloud computing has become increasingly popular in recent years due to its scalability, flexibility, and cost-effectiveness. However, cloud environments are also vulnerable to security threats such as data breaches, insider attacks, and virtual machine (VM) vulnerabilities. In a study by Rong and colleagues [23], the authors proposed a security framework for cloud environments that includes multiple security layers such as access control, data encryption, and intrusion detection/prevention. They demonstrated the effectiveness of their framework using a real-world cloud environment. Similarly, a study by Ma and colleagues [24] investigated the use of virtual machine introspection (VMI) for detecting and mitigating VM-based attacks in cloud environments.

2.2.7. Artificial Intelligence in Security.

Artificial intelligence (AI) has emerged as a powerful tool for improving network security, enabling more accurate and efficient detection and response to security threats. In a study by Zhang and colleagues [25], the authors provided a comprehensive survey of AI-based security solutions, including machine learning, deep learning, and natural language processing. They also discussed the challenges and opportunities of using AI in security, such as data privacy, explainability, and adversarial attacks. In another study, Choo and colleagues [26] proposed an AI-based approach to identifying and mitigating advanced persistent threats (APTs), a type of stealthy and long-term attack that is difficult to detect using traditional security tools.

3. Technical Overview of Proposed Solution

The problems mentioned above such as malware attacks manipulate networks and can be taken care of by identification. There are 25 common malwares used in the networks such as: Adialer.c, Dontovo.A, Skintrim.N etc.
Aritficial intelligence considered better in most of the cases where a complex identification or classification is needed. Specifically Convolutional Neural Networks(CNN) are pretty good at classifying the images or detecting objects in the image. Identification of the malwares is the first part

of our solution for malware attacks. Identification of the different malwares helps us to detect and deal with them.

CNNs are suited for our malware identification problem with only a few problems. One of them is called 'Catastrophic Forgetting'(CF). It is a common problem for neural networks and it decreases the reusability of the trained model. Catastrophic Forgetting is a tendency to forget the previously learned data on the neural network and it is not good for the sustainability of the security system of the network. For overcoming this issue there is a new method called 'Lifelong Machine Learning'(LML). It prevents the forgetting issue of the neural network.

Lifelong Learning is a learning technique that learns without interruption and accumulates the past information for adapting it to the future problems. In this learning method catastrophic forgetting is eliminated due to the rehearsal mechanism it has in the architecture of the system.

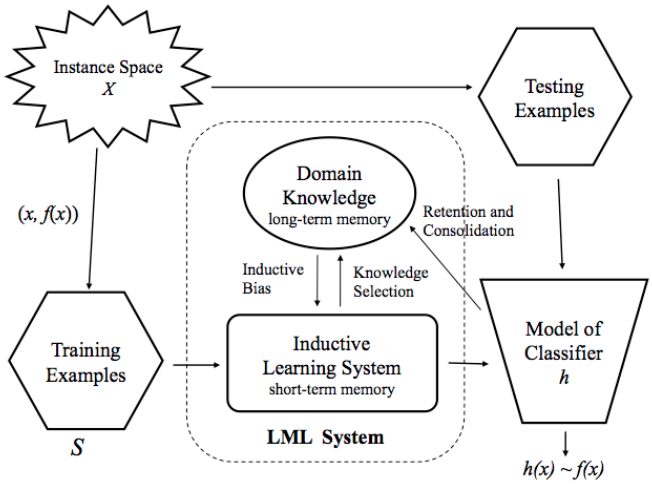

Figure 3. Lifelong Machine Learning [34]

3.1. Malware Detection

Cyber attacks in the computer networks become more and more common. Most of the viruses tend to remain silent in the network and show nearly no reaction. Their detection is nearly impossible especially in the spywares. So it is becoming more common to use computer automation methods such as Neural Networks. CNNs are known with their incredible accuracy in resolving images that comes complex to human eyes. CNNs are generally used in the object detection in other areas but they can be implemented in file classification and identification in the network security also.

There are 3 main types of learning: Unsupervised, Supervised and Reinforcement Learning. Unsupervised learning identifies from unlabeled dataset, supervised learning does the same thing from a labeled dataset and reinforcement learning learn through the interaction with the environment. [35].

Supervised learning requires a labeled dataset for training. Unsupervised learning identifies hidden data patterns from an unlabeled dataset, while Reinforcement learning does not require data as it learns by interacting with the environment.

In the researchs about malware attacks in Android systems by comparing the supervised and unsupervised learning methods, best accuracy result for malware detection is coming from the supervised learning with feature selection and the results show us feature selection improves the accuracy significantly, from 54.56% to 74.5% [36].

3.2. Comparison Between Supervised Learning Methods

Malware detection can be done in many different methods. Some of the methods used are Support Vector Machine, Decision Trees and Long Short Term Memory. In one study with over 43876 malware file data with 100 columns and 5 rows, CNN model that is combined with the LSTM performs higher accuracy(99%) than SVM(95%) and DT(98%) [37].

In some cases in order to use CNN in the different malware files, it is needed to transform the hexadecimal representation of the raw binary files to image files to 8 bit vectors and grascale images for the CNN to evaluate [38]. Differantion of the different malwares also can be identified by these grayscale images and can be reshaped according to the layers of the convolutional neural network.

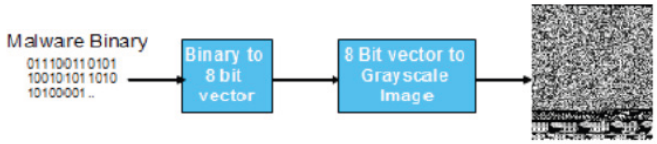

Figure 4. Converting Malware Files To Image [38]

3.3. Lifelong Machine Learning

Lifelong machine learning algorithms is like many other machine learning algorithms, try to adapting the new problems and environment, recognize patterns in the data and can predict new results according to the previously learned data. The problem with machine besides LML they tend to suffer from the performance decrease due to the loss of previously learned data. Technical cause of this problem model's weights are updated according to the new data and incompatible with the old data. This phenomenon is called catastrophic forgetting as we mentioned before in the world of artificial neural networks in machine learning.

LML eliminates the catastrophic forgetting problem, keeps its performance on the previously learned data and can use them on the new data. LML can be used with different artificial neural networks such as CNN and RNN. Different types of data can be used also in LML. One of the different types of data is graph data and it is generally used in traffic network or digital social network. One of the survey tries to

find the solution with the LML and compares it to the other forms machine learning algorithms and relizes that problems like extreme evolution where continuously different tasks of data are coming it is sometimes better to forget the old data for better prediction [39].

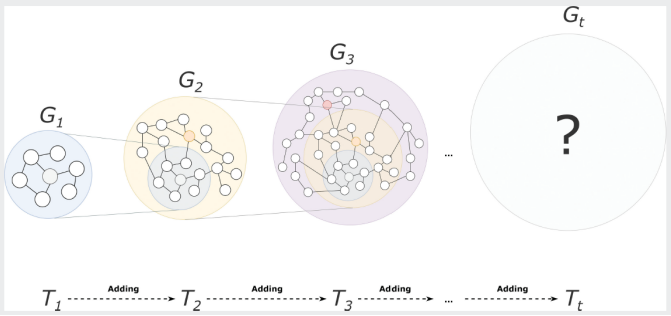

Figure 5. Incremental Development of Graph Data [39]

3.3.1. Challenges With Lifelong Machine Learning.

LML algorithms sounds good when the explanation of it mentioned but they are from problem-free still. Most common problems with the LML algorithms are generally considered as uncertain neighborhoods, global dependencies learning, extreme evolution, class imbalance and model training parallelization.

Uncertain neighborhood problem stems from where the new incoming data is very similar to the previous one so weights shouldn't be updated too much in order to prediction to be correct. Global dependencies learning problem's cause is the change of the dependency of tasks while tasks are increasing gradually. Extreme evolution problem is the opposite of the uncertain neighborhood problem where the new tasks are nearly completely different and independent from each other so changes of the weight values in the neural network should adapt better than the LML algorithm allows. Class imbalance problem arises from the unequal size of sample dataset, this can cause biased performance on certain classes where the bigger dataset sample leads to better performance while the smaller on leads to poorer. Model training parallelization problem is caused by the need of LML algorithms to be trained on multiple tasks sequentially unlike the parallelized training with GPU and CPU.

3.3.2. Elastic Weight Consolidation.

Elastic weight consolidation is one of the methods in the LML algorithms for preventing the common artificial neural network problem CF. Due to the biological similarity of human learning and LML in one article like EWC, synaptic intelligence and memory aware synaptic mass approaches can explain the metaplastic behaviour of the brain [40] but in another paper suggest that its performance even in the unnaturally stationary laboratory tasks far left behind comparing to the human's adaption against natural nonstationarity envrionments [41].

EWC is also preventing the learning from the new data harmful for the neural network. As the name suggests it updates the weights or conservates the old weights according to the situation, the key part of its prevention against CF.

This method functions by producing a penalty for the weight that are important learned tasks in the past. This is possible due to the matrix called Fisher Information Matrix (FIM) that is calculated for the weights of the previous tasks. Diagonal of the Fisher matrix is used for calculating the FIM in the process and also shows the user how valuable the which task in the past. It is then used for to produce quadratic penalty term for them. Quadratic penalty term updates the loss function used in the neural network for training and testing with the new incoming data.

Valuable weights of the previous tasks won't be changing too much to the point of forgetting due to the Importance Matrix(IM). Keeping the old information provided with the penalties of the changing old weight values. It is also important to mention that penalty term is dependent on the hyperparameter. Hyperparameter in the EWC called as regularization coefficient.

$$\boldsymbol{I}_E(\boldsymbol{\theta}) = \begin{bmatrix} \frac{n}{\sigma^2} & \frac{\sum_{i=1}^{n} z_i}{\sigma^2} & 0 \\ \frac{\sum_{i=1}^{n} z_i}{\sigma^2} & \frac{\sum_{i=1}^{n} z_i^2}{\sigma^2} & 0 \\ 0 & 0 & \frac{2n}{\sigma^2} \end{bmatrix}.$$

Figure 6. Example of Fisher Information Matrix [42]

4. Implementation

We choose to create a CNN model and implement EWC for malware classification in our project. Combination of both has not used before for the network security. Cnn needs to use data in image format so we first convert hexadecimal malware files into grayscale png images. For deciding the horizontal and vertical side of the image we used this equation.

- Length of the array of the hexadecimal file $= l$
- Length of the horizontal side $= x$
- Length of the vertical side $= y$

$$x = 2^{log_2(l*16)+1}$$

$$y = (l * 16)/x$$

Our dataset for malware images consists of 9339 images that in the shape of (64,64). Count of different type of

malware is 25. In the dataset 70% of the data is used for training and 30% of it used for test size. Percentage of each malware is listed and grayscale images of some of them are shown in the below.

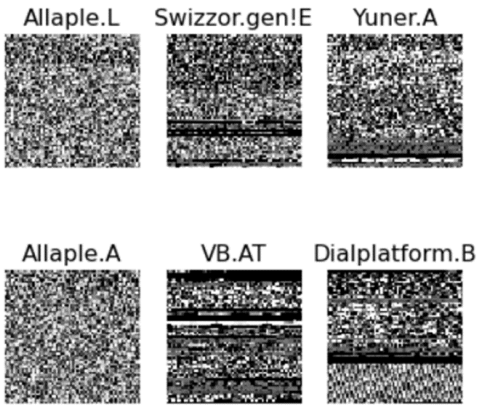

Figure 7. Malware Images

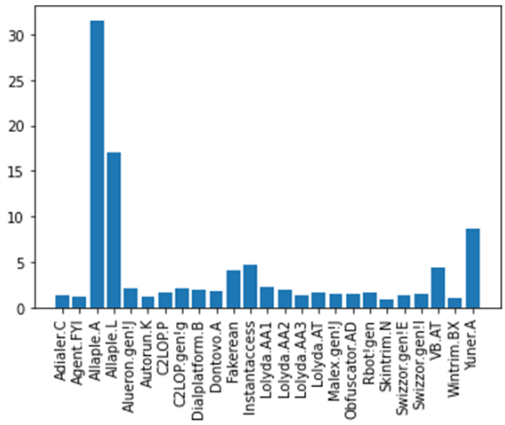

Figure 8. Malware Percentages In The Dataset

After preparing the dataset is done. We construct a sequential CNN model for training and testing.

Input layer is convolutional 2 dimensional layer with 30 filters and (3,3) kernel size. Following layers are 2 dimensional max pooling((2,2) pool), 2 dimensional convolutional(15 filter, (2,2) kernel size), max pooling((2,2) pool), dropout layer with 25% removal of neurons, flattening layer, dense layer(relu activation), dropout layer with 50% removal of neurons, dense layer(relu activation).

Output is a dense layer with 25 neurons with softmax activation. Optimizer of the model is "Adam" and loss function of the neural network is "Categorical Cross Entropy".

Model's output function will be changed according to the FIM, IM and Regularization Terms(RT) in the final evaluation and comparison for a CNN with and without EWC at the end.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 30)        840

 max_pooling2d (MaxPooling2D  (None, 31, 31, 30)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 15)        4065

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 15)       0
 2D)

 dropout (Dropout)           (None, 14, 14, 15)        0

 flatten (Flatten)           (None, 2940)              0

 dense (Dense)               (None, 128)               376448

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 50)                6450

 dense_2 (Dense)             (None, 25)                1275
...
Total params: 389,078
Trainable params: 389,078
Non-trainable params: 0
_________________________________________________________________
```

Figure 9. Model Summary

Before implementing new EWC, we test the model with new dataset with the same size for examining the forgetting.After the test results we designed a new loss function with ewc for test again and steps of it is shown in the below:

1) Storing the old weights of the model.
2) Determining the regularization strength parameter, $\lambda = 0.5$.
3) Calculating the FIM for each weight in Conv2D adn Dense layers.
4) Calculating the IM with FIM values and $lambda$.
5) Calculating RT for each weight.
6) Calculating new weights with FIM, IM and λ. Calculation of it will be shown in the below.
7) Updating the loss function.
8) Using the new loss function on the previously trained model.

After storing the weights and determining the λ, Computation of FIM comes next. Algorithm is shown step by step:

1) After stroing the old weights for updating them at the end and determining the regularization of strength parameter, initializing FIM with zeros comes next.

```
fisher = []
for v in range(len(var_list)):
    fisher.append(np.zeros(var_list[v]))
```

2) Computing gradients for each weight.

```
for data in dataset:
    inputs, labels = data
    outputs = model(inputs)
    loss = loss_func(outputs, labels)
    grad = tf.gradients(loss_fn(model.
         output, y_train), weight)
```

3) Square of the gradients and are added.

```
for i, param in enumerate(weights):
    fisher_matrix[i] += (grad[i] ** 2)
```

4) Normalizing FIM with number of samples.

```
num_samples = len(dataset)
for i in range(len(fisher_matrix)):
    fisher_matrix[i] /= num_samples
```

Next step is computation of IM. Algorithm is shown below:

```
for fisher_matrix in fisher_matrices:
    IM = lambda * (fisher_matrix /
         previous_task_accuracy)
    IM += (1 - lambda) * (fisher_matrix /
         new_task_accuracy)
    IM.append(IM)
```

Calculation of RT will be possible with our IM now.

```
for i, layer in enumerate(layers):
    weights = layer.get_weights()
    RT=lambda * matmul(IM[i],(weights**2))
    RT.append(RT)
```

After the computation of the RT is done we can change the loss funtion with it and train and test the new model.

```
ewc_loss = categorcal_cross_entropy
for regularization_term in RT:
    ewc_loss += regularization_term
```

Now The results must show us that previous tasks' accuracy shouldn't suffer due to forgetting and new tasks' accuracy must be higher than our testing. Also loss values must be improved on the new training.

5. Analysis of Implementation

In our first training set 6537 labeled image is used and 2802 labeled image used for testing. Batch size is 32, epoch count is 25 and shapes of the images are 64 x 64. Same size of training and testing data is used for new set of data also. Loss and accuracy values before EWC implementation are shown in the below.

TaskName	*Accuracy*	*Loss*
First Dataset	0.9518	0.156
Second Dataset	0.9704	0.097

After the implementation of the new loss function with implementation with EWC, Results met our expectations about the task 1 but there is no improvement on the task 2 as we see in the below.

TaskName	*Accuracy*	*Loss*
First Dataset	0.9793	0.07
Second Dataset	0.9704	0.097

Changes on the loss and accuracy on the tasks are:

TaskName	*Accuracy*	*Loss*
First Dataset	+0.0275	-0.086
Second Dataset	0.0	0.0

Test results showed that EWC implementation prevented CF but didn't improve the new task's accuracy with the previous knowledge. At the end our research showed that LML method, EWC can be used in the malware detection systems for longer time than their counterparts due to its CF prevention thus LML approach also maintains longer lifetime which is more sustainable then without.

6. Conclusion

The goal of this research paper is to explore the use of LML algorithms for network attacks to extend the lifespan of network security systems. We present a review of the literature on LML algorithms for network attacks and discuss different approaches to implementing these algorithms. We also present a case study on the use of LML algorithms for network intrusion detection, demonstrating how they can improve the accuracy and effectiveness of intrusion detection systems. We highlight the challenges and limitations of using LML algorithms for network security and propose directions for future research. Our findings suggest that LML algorithms have significant potential for extending the lifespan of network security systems. By continuously learning and adapting to new data, these algorithms can help to improve the accuracy and effectiveness of network security measures, and enable systems to evolve and adapt over time to the changing threat landscape. Overall, this research paper contributes to the growing body of knowledge on the use of LML algorithms for network security and highlights their potential to enhance the lifespan of security systems. This paper is of interest to researchers and practitioners in the field of network security who are interested in exploring innovative approaches to address the ever-evolving threat landscape. Lifelong machine learning algorithms for network attacks have certain potential advantages, but they also have some drawbacks. Future research can concentrate on creating algorithms that use less data and are more efficient, as well as on enhancing interpretability via explainable AI methodologies. To fully grasp the promise of these algorithms, it will also be crucial to assess their performance in real-world circumstances.

References

[1] H. Yuan, Y. Tang, W. Sun, and L. Liu, "A detection method for android application security based on tf-idf and machine learning," PLoS One, vol. 15, no. 9, pp. e0238361, Sep. 2020.

[2] C. S. Yadav et al., "Malware Analysis in IoT & Android Systems with Defensive Mechanism," Electronics, vol. 11, no. 15, pp. 2354, Jul. 2022.

[3] M. Arse, K. Sharma, S. Bindewari, A. Tomar, H. Patil and N. Jha, "Mitigating Malware Attacks using Machine Learning: A Review," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, 2023, pp. 1032-1038, doi: 10.1109/AISC56616.2023.10085630.

[4] S. Yan et al., "A Survey of Adversarial Attack and Defense Methods for Malware Classification in Cyber Security," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 467-496, Firstquarter 2023, doi: 10.1109/COMST.2022.3225137.

[5] S. Wang, "Intelligent Algorithm in Computer Network Security," 2022 International Conference on Knowledge Engineering and Communication Systems (ICKES), Chickballapur, India, 2022, pp. 1-4, doi: 10.1109/ICKECS56523.2022.10059831.

[6] Z. Chen and B. Liu, Lifelong Machine Learning, 2nd ed. Morgan & Claypool Publishers, 2020.

[7] O. Gheibi and D. Weyns, "Lifelong Self-Adaptation: Self-Adaptation Meets Lifelong Machine Learning," 2022 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Pittsburgh, PA, USA, 2022, pp. 1-12, doi: 10.1145/3524844.3528052.

[8] Wang, S., Tao, X., Yang, P., & Zhang, Y. (2021). A Survey on Lifelong Machine Learning: From a Machine Learning Perspective. IEEE Access, 9, 28924-28944. [6]S. Wang, "Intelligent Algorithm in Computer Network Security," 2022 International Conference on Knowledge Engineering and Communication Systems (ICKES), Chickballapur, India, 2022, pp. 1-4, doi: 10.1109/ICKECS56523.2022.10059831.

[9] D. Lemos, "Code-Red Worm Causing Concern," *eWeek*, 2001. [*Online*]. Available: https://www.eweek.com/security/code-red-worm-causing-concern. [Accessed: Apr. 4, 2023].

[10] N. Perlroth, "Equifax Says Cyberattack May Have Affected 143 Million in the U.S.," *The New York Times*, 2017. [*Online*]. Available: https://www.nytimes.com/2017/09/07/business/equifax-cyberattack.html. [Accessed: May 2, 2023].

[11] J. Abu and J. Doe, "Third-Party Security Assessments and Supply Chain Risk Management," *Journal of Cybersecurity*, vol. 12, no. 3, pp. 207–222, 2020.

[12] P. Roberts, "WannaCry Ransomware: Everything You Need to Know," *The Guardian*, 2017. [*Online*]. Available: https://www.theguardian.com/technology/2017/may/15/wannacry-ransomware-everything-you-need-to-know. [Accessed: May 2, 2023].

[13] R. Ronen, A. Thompson, A. Taivalkoski, A. Haghighatkhah, T. Reynolds, and M. Almgren, "The SolarWinds Supply Chain Attack: What You Need to Know," *Digital Shadows*, 2021. [*Online*]. Available: https://www.digitalshadows.com/blog-and-research/the-solarwinds-supply-chain-attack-what-you-need-to-know/. [Accessed: May 2, 2023].

[14] D. E. Sanger, N. Popper, and E. Corsi, "Cyberattack Forces a Shutdown of a Top U.S. Pipeline," *The New York Times*, 2021. [*Online*]. Available: https://www.nytimes.com/2021/05/08/us/politics/cyberattack-colonial-pipeline.html. [Accessed: May 2, 2023].

[15] H. Lu, Y. Yang, J. H. Park, M.H. Dehkordi, and S.Rho, "Ddos attack detection and mitigation using machine learning in software-defined networks," *IEEE Access*, vol. 7, pp. 10,423–10,433, 2019.

[16] M. Kim, Y. Park, Y. Kim, and H. Kim, "Lightweight defense against DoS attacks on IoT devices using blockchain," *IEEE Access*, vol. 8, pp. 53,754–53,765, 2020.

[17] S. Khan, N. K. Ali, and T. Khan, "Phishing detection using machine learning and feature engineering," *IEEE Access*, vol. 8, pp. 36,625–36,637, 2020.

[18] N. Abdelhamid, I. Ayuba, and A. T. Abdelhamid, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, Article e01802, 2019. [*Online*]. Available: https://doi.org/10.1016/j.heliyon.2019.e01802. [Accessed: May 2, 2023].

[19] A. Alzahrani, F. Alajlan, and M. S. Al-Rodhaan, "Network intrusion detection using deep learning: A review," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2103–2135, 2020.

[20] A. Bahga and V. K. Madisetti, *Blockchain for distributed systems security*. Springer International Publishing, 2020.

[21] E. Ronen, D. Falkner, R. Teixeira, and N. Nemitz, "Towards a security assessment framework for blockchain systems," *IEEE Security & Privacy*, vol. 18, no. 4, pp. 48–57, 2020.

[22] A. Ali, A. Aburrous, I. Awan, and M. A. Khan, "A blockchain-based secure communication architecture for IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9086–9098, 2020.

[23] W. Rong, C. Li, and Y. Wang, "A cloud security framework based on service-oriented architecture," *IEEE Access*, vol. 8, pp. 59,468–59,480, 2020.

[24] X. Ma, S. Kang, S. Kim, H. K. Kim, and J. Kim, "A virtual machine introspection based security solution for cloud computing environments," *IEEE Access*, vol. 8, pp. 53,818–53,830, 2020.

[25] Z. Zhang, W. Yao, Y. Li, and W. Yang, "Artificial intelligence for cybersecurity: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2597–2633, 2019.

[26] K.-K. R. Choo, C. L. Yang, and H. Y. Lin, "An intelligent approach for advanced persistent threat detection and mitigation," *IEEE Access*, vol. 8, pp. 35,159–35,168, 2020.

[27] Sharma, A., Sharma, P., & Kumar, R. (2021). Deep learning-based approach for malware detection using static and dynamic analysis. *Journal of Ambient Intelligence and Humanized Computing*, 1-14.

[28] Gupta, S., Bhat, S. K., & Chauhan, R. (2021). Machine learning-based malware classification framework for behavior analysis. *Computers, Materials & Continua*, 69(1), 847-865.

[29] Shafique, M. F., Ali, S., Rehman, Z. U., & Raza, B. (2021). A comprehensive survey on wireless security: Issues, challenges and solutions. *Journal of Network and Computer Applications*, 180, 102982.

[30] Li, Y., Guo, Y., Liu, J., & Shi, J. (2021). An SDN-based approach to enhance wireless security with integrated security functions. *IEEE Access*, 9, 67846-67856.

[31] Popper, N., Conger, K., & Alba, D. (2020, July 16). Twitter Struggles to Unpack a Hack Within Its Walls. The New York Times. Retrieved from https://www.nytimes.com/2020/07/16/technology/twitter-hack-investigation.html

[32] Goodin, D. (2021, March 2). Tens of thousands of US organizations hit in ongoing Microsoft Exchange hack. Ars Technica. Retrieved from https://arstechnica.com/information-technology/2021/03/tens-of-thousands-of-us-organizations-hit-in-ongoing-microsoft-exchange-hack/

[33] Barth, B., & Dwoskin, E. (2021, July 5). Cyberattack on Florida technology firm hits 200 U.S. businesses, suspected Russian group demands $70 million. The Washington Post. Retrieved from https://www.washingtonpost.com/business/2021/07/05/kaseya-cyberattack/

[34] Consolidation Using Sweep Task Rehearsal: Overcoming the Stability-Plasticity Problem - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/A-framework-for-Lifelong-Machine-Learning_fig1_277871832 [accessed 23 Apr, 2023]

[35] R. Khandelwal, "Supervised, unsupervised, and reinforcement learning," Medium, 20-Jul-2022. [Online]. Available: https://arshren.medium.com/supervised-unsupervised-and-reinforcement-learning-245b59709f68. [Accessed: 04-May-2023].

[36] L. Meijin, F. Zhiyang, W. Junfeng, C. Luyu, Z. Qi, Y. Tao, W. Yinwei, and G. Jiaxuan, "A systematic overview of Android malware detection," Applied Artificial Intelligence, vol. 36, no. 1, 2021.

[37] M. S. Akhtar and T. Feng, "Detection of malware by Deep Learning as CNN-LSTM machine learning techniques in Real time," Symmetry, vol. 14, no. 11, p. 2308, 2022.

[38] J. M. Waghmare and M. M. Chitmogrekar, "A review on malware detection methods," SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology, vol. 14, no. 01, pp. 38–43, 2022.

[39] F. G. Febrinanto, F. Xia, K. Moore, C. Thapa, and C. Aggarwal, "Graph lifelong learning: A survey," IEEE Computational Intelligence Magazine, vol. 18, no. 1, pp. 32–51, 2023.

[40] P. Jedlicka, M. Tomko, A. Robins, and W. C. Abraham, "Contributions by metaplasticity to solving the catastrophic forgetting problem," Trends in Neurosciences, vol. 45, no. 9, pp. 656–666, 2022.

[41] S. Pisupati and Y. Niv, "The challenges of lifelong learning in biological and artificial systems," Trends in Cognitive Sciences, vol. 26, no. 12, pp. 1051–1053, 2022.

[42] ThomasThomas 17311 silver badge99 bronze badges and Travis C CuvelierTravis C Cuvelier 57933 silver badges1010 bronze badges, "Fisher information matrix for normal distribution," Mathematics Stack Exchange, 01-Feb-1966. [Online]. Available: https://math.stackexchange.com/questions/3219926/fisher-information-matrix-for-normal-distribution. [Accessed: 29-Apr-2023].