PHÁT HIỆN VÀ PHÂN LOẠI MÃ ĐỘC TỐNG TIỀN DỰA TRÊN HÌNH ẢNH SỬ DỤNG PHƯƠNG PHÁP HỌC SÂU

IMAGE-BASED DEEP LEARNING APPROACH FOR DETECTION AND CLASSIFICATION OF RANSOMWARE

SVTH: Đàm Quang Tiến, Nguyễn Nghĩa Thịnh, Lê Viết Trung

Lớp 18T1 và 18TCLC-DT3, Khoa Công nghệ Thông tin, Trường Đại học Bách Khoa – Đại học Đà Nẵng;
Email: damtien440@gmail.com, nghiathinh2000@gmail.com, trunglv2000vn@gmail.com

GVHD: TS. Lê Trần Đức

Khoa Công nghệ Thông tin, Trường Đại học Bách Khoa – Đại học Đà Nẵng; Email: letranduc@dut.udn.vn

Tóm tắt

Các mã độc tống tiền ngày càng phát triển nhanh và đi kèm với nhiều kĩ thuật để qua mặt các phương pháp phát hiện cổ điển. Trong nghiên cứu này, đề xuất một phương pháp mã hóa các các thông tin đặc trưng từ PE headers vào hình ảnh trực quan của ransomware. Phương pháp này bao gồm việc sử dụng machine learning để lựa chọn ra các đặc trưng có tác động lớn đến phân loại của mẫu dữ liệu, sau đó mã hóa theo các quy tắc màu sắc vào bức ảnh. Dựa vào đó xây dưng một mô hình kết hợp các mô hình deep learning như VGG16 và ResNet50. Cùng với nhiều thực nghiệm, phương pháp của chúng tôi đã đạt được kết quả cao, trong đó lên tới 99.85% trong thử nghiệm phân loại ransomware và benign.

Từ khóa – Ransomware; Deep learning; Machine Learning; Ensemble learning; Image-based diagnose; PE header

Abstract

Recently, ransomware is evolving rapidly and comes with many techniques to bypass classical detection malware methods. In this study, we proposed a method to encode the specific information from PE headers into the ransomware's visual image. This method consists of using machine learning to select features that have a great impact on the classification of data sample, and then encoding according to color rules into the image. Based on that, we build a model that combines deep learning models such as VGG16 and ResNet50. Through with experiments, our method has achieved high results, in which up to 99.85% in the test classifying ransomware and benign.

Key words – Ransomware; Deep learning; Machine Learning; Ensemble learning; Image-based diagnose; PE header

1. Giới thiệu

Hiện nay malicious software (phần mềm độc hại, gọi tắt là Malware) đã trở thành mối đe dọa lớn và là công cụ đắc lực trong các cuộc tấn công mạng. Trong những loại mã độc phổ biến đang hoạt động, mã độc tống tiền (ransomware) nổi lên như một dạng mã độc nguy hiểm và gây ra thiệt hại rất lớn cho các cá nhân, công ty, doanh nghiệp nói riêng và cho nền kinh tế nói chung. Điều đáng nói mã độc hiện đại ngày càng tinh vi, sử dụng nhiều kỹ thuật lẩn trốn, duy trì sự tồn tại của chúng trên hệ thống máy tính. Chính vì thế các phương pháp phân tích mã độc cũ gần như không thể nào theo kịp với những đợt tấn công mới và các biến thể mới. Quá trình phân tích mã độc thường dựa nhiều vào kinh nghiệm cũng như kiến thức của người thực hiện do đó các kết quả phân tích bị hạn chế đặc biệt là khi phân tích những mẫu mã độc mới. Từ đó có thể thấy cần phải có những giải pháp, kỹ thuật mới hỗ trợ quá trình phân tích mã độc này.

Những năm gần đây, việc ứng dụng các thuật toán học máy và trí tuệ nhân tạo đã trở nên phổ biến trong rất nhiều lĩnh vực. Không nằm ngoài xu thế đó, áp dụng trí tuệ nhân tạo mà cụ thể là học sâu (deep learning) và mạng nơron (neural network) để giải quyết các vấn đề liên quan đến an toàn thông tin mạng nói chung, phân tích mã độc nói riêng là một bước đi tất yếu. Việc tiền xử lý dữ liệu sẽ giúp các mô hình neural network này thích ứng với bài toán phân loại và nhận biết ransomware so với các tập tin hệ thống thông thường trong hệ điều hành Windows hay cụ thể là các tập tin thực thi Portable Executable (PE).

Trong nghiên cứu này, nhóm tác giả muốn thực hiện phân loại và phát hiện các ransomware dựa trên phương pháp tiền xử lý dữ liệu thành hình ảnh có chứa các đặc trưng ngữ nghĩa của các PE headers, và xây dựng một mô hình deep learning kết hợp để học được nhiều đặc trưng hơn từ dữ liệu hình ảnh nhằm tăng độ chính xác của kết quả so với các thuật toán hiện có. Nhờ đó đóng góp một phương pháp mới cho lĩnh vực trong quá trình chống lại các loại mã độc tống tiền.

Những đóng góp chính trong nghiên cứu này là:

(1) Phân loại ransomware dùng hình ảnh có chứa thông tin từ PE headers, từ đó giúp tăng tính tương đồng giữa các mẫu ở cùng chủng biến thể.
(2) Lựa chọn các đặc trưng từ PE headers sử dụng các mô hình học máy (Machine learning) và mã hóa vào hình ảnh thể hiện mẫu ransomware.
(3) Xây dựng mô hình kết hợp từ các mạng CNN nổi tiếng như ResNet-50, VGG16. Sử dụng mô hình mới nổi VisionTransformer vào trong bài toán phân loại mã độc tống tiền.

Phần còn lại của báo cáo này được sắp xếp như sau. Phần 2 là các nghiên cứu liên quan. Phương pháp được đề xuất trong nghiên cứu được đề cập tại phần 3. Phần 4 thể hiện kết quả thực nghiệm của phương pháp. Cuối cùng, phần 5 tổng hợp về toàn bộ nghiên cứu.

2. Các nghiên cứu liên quan

2.1. Phát hiện và phân loại mã độc dựa trên hình ảnh khác

(Nataraj, et al., 2011) đề xuất một phương pháp trực quan hóa mã độc chuyển từ mã nhị phân sang hình ảnh bằng cách ánh xạ đơn giản từng cụm 8bit nhị phân thành 1pixel thể hiện trên hình ảnh. Phương pháp này hoạt động đơn giản và từng đạt được hiệu quả cao và đáng tin cậy.

Tuy nhiên, khi các loại mã độc đã bị làm nhiễu hay các thủ thuật làm che dấu thì chỉ ánh xạ trực tiếp từng pixel là không đủ để nâng cao độ chính xác.

(Daniel, et al., 2020) đưa ra những khảo sát cho thấy sự ảnh hưởng và mức độ quan trọng của các phương pháp phân tích tĩnh đặc biệt là sử dụng PE headers cho mục đích phân loại hay xác định malware. Thông qua bài khảo sát, chúng tôi có thêm cơ sở để ứng dụng machine learning để phân tích và trích xuất PE headers trong mục tiêu phân loại và xác định malware.

2.2. Các phương pháp xử lý ảnh dựa trên deep learning

Xử lý ảnh sử dụng Deep learning là một trong những bài toán được ứng dụng rất nhiều trong thực tiễn như nhận diện khuôn mặt, phân loại các vật thể… Đi theo đó thì dần có nhiều mô hình mới ra đời để xử lý các bài toán phức tạp hơn như nhận diện chữ cái viết tay, trích xuất thông tin từ hồ sơ xin việc, chứng minh nhân dân… Với những lợi ích to lớn mà xử lý ảnh mang lại đã đem đến nhiều giải pháp tối ưu thay thế cho các bài toán truyền thống, trong đó có bài toán phát hiện và phân loại các loại mã độc.

Trước đây, một số phương pháp Machine learning đã được sử dụng để phân loại mã độc sau khi chuyển sang dạng ảnh Gray hoặc RGB chứa các thông tin đặc trưng của các loại mã độc và dựa vào các thuật toán ML như Kmean, SVM để phân loại và đạt được độ chính xác cao 95% với bộ dữ liệu chứa 25000 malware và 12000 benign (Kancherla & Mukkamala, 2013). Tuy nhiên với sự phức tạp cũng như sự gia tăng về số lượng không ngừng của mã độc thì các phương pháp ML này không đáp ứng được về độ chính xác. Image-based Deep Learning là chính giải pháp, bằng phương pháp end-to-end được ứng dụng rất nhiều trong các bài toán phân loại nói chung với các base model CNN như ResNet, VGG, EfficientNet, ... được huấn luyện trên tập ImageNet với hơn 10 triệu ảnh khác nhau và cho ra độ chính xác rất cao. Để áp dụng Image-based Deep Learning vào bài toán phát hiện và phân loại mã độc đặc biệt là loại mã độc Ransome sẽ có hai trường hợp chính.

(1) Nếu dữ liệu mã độc thu thập đã được đánh nhãn có số lượng rất lớn (tầm vài triệu ảnh, ...) thì ta sẽ huấn luyện toàn bộ dữ liệu bằng toàn bộ các parameter của các mô hình CNN.

(2) Nếu dữ liệu mã độc thu thập đã được đánh nhãn có số lượng rất lớn (tầm vài triệu ảnh, ...) thì ta sẽ huấn luyện toàn bộ dữ liệu bằng toàn bộ các parameter của các mô hình CNN. Nếu dữ liệu mã độc thu thập đã được đánh nhãn có số lượng ít (tầm vài nghìn mẫu, ...) thì giải pháp là sử dụng phương pháp Transfer learning được huấn luyện trên tập dữ liệu lớn (ImageNet, COCO…).

Trong lần nghiên cứu này, chúng tôi thực hiện huấn luyện theo phương pháp thứ 2 vì việc thu thập dữ liệu các loại mã độc đặc biệt là mã độc Ransom là rất khó khăn nên số lượng dữ liệu thu được là không nhiều (Kalita, et al., 2020) đã chứng minh điều này, khi sử dụng CNN + SVM đối với mô hình Resnet và Vgg16 khi không dùng transfer learning cho ra acc lần lượt là 26,66% và 14,31%.

Đã có nhiều nghiên cứu ứng dụng phương pháp end-to-end bằng Transfer Learning cho bài toán trên với 90.77% cho VGG16 + Softmax, 98.62% cho ResNet + Softmax (Rezende, et al., 2017).

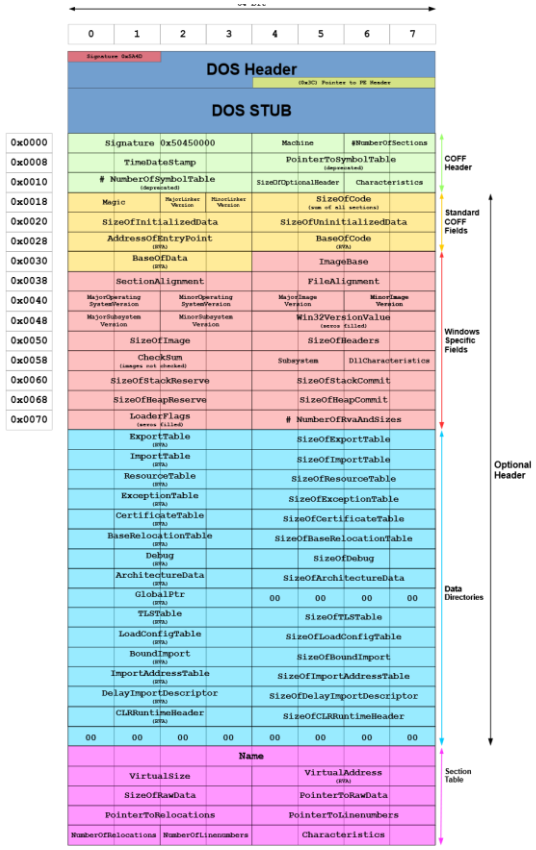
Tuy nhiên với sự phát triển không ngừng của các loại mã độc như làm rối (obfuscation), nén (packed) làm ảnh hưởng đáng kể đến độ chính xác của bài toán. Điều này yêu cầu mô hình có độ phức tạp cao hơn, hoặc là một giải pháp khác chỉ tập trung vào đặc trưng tốt nhất của loại mã độc đó. Để giải quyết bài toán đầu tiên thì nhóm dùng phương pháp Ensemble learning bằng cách kết hợp mô hình Resnet và Vgg16 và bài toán ML truyền thống lại với nhau. Ngoài ra với mô hình Vision Transformer theo cơ chế Multihead Attention tập trung vào những đặc trưng tốt nhất (the best features) để giải quyết bài toán thứ hai.

3. Phương pháp

3.1. Chọn ra đặc trưng tốt từ PE headers sử dụng ML

Portable Executable (PE) là một loại định dạng tệp thực thi, mã đối tượng, định dạng đuôi DLL và những thứ khác được sử dụng trong phiên bản 32-bit và 64-bit của HĐH Windows. PE có nguồn gốc từ đặc tả Định dạng tệp đối tượng chung (COFF - Common Object File Format), cũng được sử dụng bởi hầu hết các tệp thực thi Unix. Cái tên portable xuất phát từ thực tế là định dạng không dành riêng cho kiến trúc nào cả.

PE headers chứa thông tin, được đọc bởi window loader khi chúng ta thực thi mã nhị phân. Sau đó, nội dung nhị phân sẽ được tải từ tệp vào bộ nhớ. Tệp PE là một cách chương trình thông báo cho HĐH về các yêu cầu thực thi của nó vì nó chỉ ra nơi tệp thực thi cần được tải vào bộ nhớ. Việc kiểm tra các PE Headers mang lại nhiều thông tin về hệ nhị phân hữu ích và các chức năng của nó. Do đó, PE Headers có tầm quan trọng lớn trong việc phát hiện và phân tích phần mềm độc hại.



Hình 1. *Cấu trúc của một PE headers*

Thông qua đánh giá và khảo sát ý nghĩa thuộc tính của PE headers, có rất nhiều thuộc tính mang ý nghĩa categorical. Ví dụ như feature Machine phần lớn sẽ chỉ có 2 giá trị là 332, 34404 hay SizeOfOptionalHeader cũng hầu hết sẽ có 1 trong 2 giá trị là 224, 240. Các giá trị có thể có xu hướng tăng dần nhưng về bản chất đều mang ý nghĩa phân loại. Từ những kết luận trên, chúng tôi quyết định lựa chọn những mô hình ML làm việc tốt với loại dữ liệu categorical, đặc biệt chúng tôi cố gắng thử nghiệm các loại mô hình hiện đại lúc bấy giờ như ExtraTrees, XGBoost và CatBoost để tăng khả năng đánh giá các feature quan trọng.

Việc chúng tôi lựa chọn những mô hình này chỉ đơn thuần vì các mô hình hoạt động tốt trên các loại dữ liệu mang ý nghĩa categorical, đặc biệt đây là những mô hình đã đạt giải thưởng trong các cuộc thi về Machine Learning do Kaggle tổ chức.

Bảng 1. *Mức độ tác động của các đặc trưng trong PE headers ảnh hưởng đến kết quả của mô hình ML*

	Features	Impact level (%)
1	'CheckSum'	11.56
2	'SizeOfUninitializedData'	10.63
3	'SizeOfStackCommit'	7.84
4	'DllCharacteristics'	7.55
5	'MinorLinkerVersion'	7.31
6	'SectionAlignment'	5.45
7	'SectionMaxRawsize'	4.83
8	'ImportsNbDLL'	4.76
9	'ImageBase'	4.61
10	'SectionsMinVirtualsize'	4.09
11	'SizeOfHeaders'	3.93
12	'MinorOperatingSystemVersion'	3.01

Với mỗi dataset khác nhau, chúng tôi sẽ tiến hành train lại để trích xuất đặc trưng. Với mỗi lần train, đầu tiên chúng tôi đặt mục tiêu độ chính xác chung của tất cả mô hình được lựa chọn để train (ví dụ với bài toán phân loại ransomeware và malware chúng tôi đạt độ chính xác chung là 99.6%). Sau đó, chúng tôi tiến hành train trên mỗi model với mục tiêu đạt được độ chính xác chung đủ 3 lần trên mỗi model. Để tăng xác suất đạt được độ chính xác chung, mỗi lần train chúng tôi sẽ random lại bộ dữ liệu train và test. Sau đó với mỗi model này chúng tôi mới tiến hành trích xuất best feature và mức độ ảnh hưởng của mỗi feature. Cuối cùng để lựa chọn nhanh feature, chúng tôi lấy trung bình cộng giá trị ảnh hưởng của mỗi feature trên tất cả model và tiến hành lựa chọn những feature ảnh hưởng nhất.

3.2. Sinh hình ảnh thang xám đại diện mẫu kết hợp thông tin PE headers

3.2.1. Cảm hứng

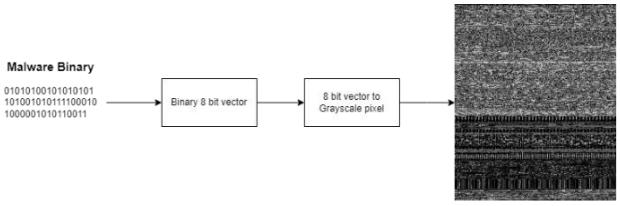
(Xiao, et al., 2021) đã phát triển một phương pháp sử dụng image-based trên đó có nhúng thông tin của các vạch phân vùng vào bức ảnh, các thử nghiệm của họ đã chứng minh được việc thêm thông tin vạch phân vùng sẽ giúp các tác vụ phân loại được diễn ra với độ chính xác cao hơn với một mô hình deep learning đơn giản.

Trong các tác vụ phân tích mã độc trên hệ điều hành window PE headers đóng vai trò quan trọng trong các nhiệm vụ phân tích tĩnh, chúng tôi nảy lên ý tưởng sử dụng thêm các thông tin khai thác được từ PE headers để cài đặt vào trong ảnh đại diện của mẫu dữ liệu đó, từ đó có thể tăng độ chính xác khi sử dụng các phương pháp phân loại sử dụng deep learning.

Để thực hiện được công việc trên, có một số việc được chúng tôi tiến hành, bao gồm, lựa chọn ra những features từ PE headers, mã hóa dữ liệu của các features đó và dựng lại hình ảnh đại diện cho mẫu phân tích.

3.2.2. Từ nhị phân sang hình ảnh

(Nataraj, et al., 2011) đã chuyển đổi từng byte nhị phân thành một pixel ảnh, được mô tả ở **Hình 2**. Các mẫu có xuất phát chung một nhánh biến thể thì có các đặc trưng phân bố tương tự nhau, tuy vậy do các kĩ thuật làm rối khiến thể hiện phân bố byte nhị phân không còn đại diện được cho biến thể mã độc nữa nên kết quả của phương pháp naive này sẽ nhanh chóng bị qua mặt do đó có nhiều phương pháp sử dụng phương pháp này như một bước trung gian để tạo nên hình ảnh đại diện cho mẫu dữ liệu. Độ rộng của bức ảnh được quy định như bảng **Bảng 2**.



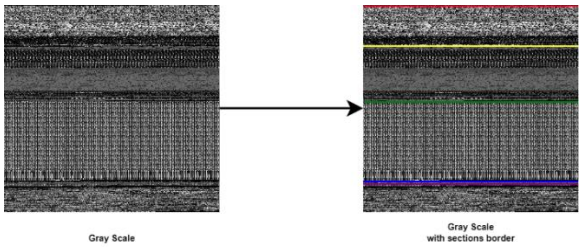
Hình 2. *Trực quan hóa mẫu dữ liệu bằng hình ảnh thang xám*

***Bảng 2**. Độ rộng hình ảnh với mỗi kích thước tệp tin.*

File Size Range	Image Width
<10 kB	32
10 kB – 30 kB	64
30 kB – 60 kB	128
60 kB – 100 kB	256
100 kB – 200 kB	384
200 kB – 500 kB	512
500 kB – 1000 kB	768
>1000 kB	1028

3.2.3. Thêm đường đánh dấu section

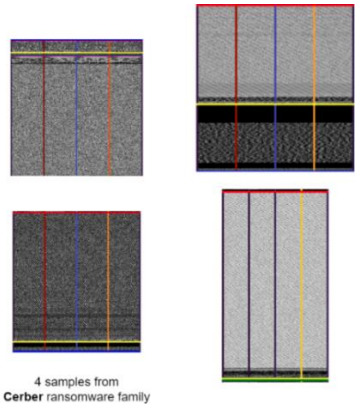
Trong phần này chúng tôi thực hiện lại ý tưởng của (Xiao, et al., 2021) được mô tả ngắn gọn như sau. Các tệp tin thực thi của hệ điều hành Window đều có một phân vùng quan trọng chứa các thông tin hướng dẫn hệ điều hành thực thi các đoạn mã chứa trong chương trình. Các thông tin về sections chứa trong PE headers bao gồm 3 thông tin cần quan tâm, đó là con trỏ đến vị trí bắt đầu section, kích thước section và tên của section. Bằng việc thêm các thông tin của sections giúp phân định rõ ràng các khu vực chứa các nội dung sẽ giúp hiểu rõ được đó là một mẫu phân tích đến từ biến thể nào. Các bước mô phỏng việc thêm thông tin sections được mô tả ở **Hình 3.** Với mỗi sections có trong PE headers sẽ tương ứng với mỗi đường thẳng nằm ngang sử dụng phương pháp mã hóa màu theo một scheme màu cố định theo tên của các phần. Độ dày của đường thẳng nằm ngang có tỉ lệ tuyến tính với độ rộng của hình ảnh.



***Hình 3.** Thêm các vạch section cho một tệp tin PE điển hình*

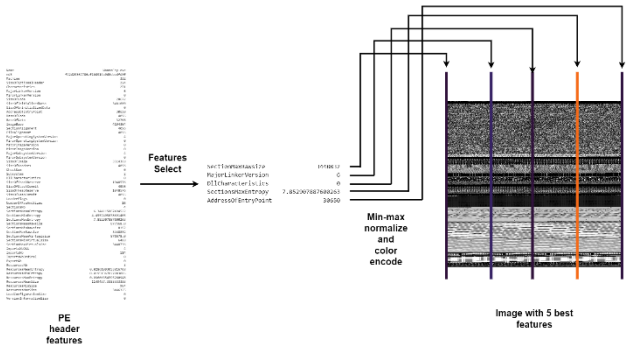
3.2.4. Mã hóa đặc trưng PE header vào hình ảnh

Trong PE headers, chứa các thông tin quy định và hướng dẫn hệ thống thực thi chương trình các thông tin có được từ đó là các thông tin quan trọng có thể sử dụng để phân biệt/phân loại các tệp tin thực thi PE. Số lượng đặc trưng nằm trong PE headers lên đến hơn 50 đặc trưng, việc mã hóa toàn bộ chúng vào là không cần thiết. Chúng tôi tuyển chọn ra những đặc trưng mà chúng tôi đánh giá mức độ ảnh hưởng là cao nhất để đặt vào trong hình ảnh đại diện mẫu. Triển khai các mô hình ML trên tập dữ liệu để thực hiện tác vụ phân biệt, từ đó có thể lựa chọn ra những đặc trưng ảnh hưởng nhất đến quyết định phân loại của mô hình ML đó. Bằng việc lựa chọn như vậy, các đặc trưng được đưa vào hình ảnh đại diện mẫu sẽ là những đặc trưng chứa những thông tin giá trị nhất cho các tác vụ phân loại. Số lượng của các feature này sẽ phụ thuộc vào loại mô hình, loại dữ liệu mà sử dụng cho phù hợp, do đó khi muốn sử dụng phải tuning hyper-parameter này.



***Hình 4**. Mẫu đến từ cùng một chủng biến thể mà việc thêm thông tin PE headers là cần thiết*

Để mã hóa các giá trị của các đặc trưng thuộc PE headers đó, thực hiện chuẩn hóa min-max cho các giá trị của tất các các đặc trưng và sau đó ánh xạ vào các giá trị từ $0 - 255$. Từ đó chọn ra màu sắc đại diện cho giá trị đó từ bảng màu Turbo Color Scheme (Blog Google AI, 2019). Việc thêm các thông tin đặc trưng của các mẫu dữ liệu có thể giúp tăng đáng kể độ chính xác của các mô hình nhận biết bởi vì các đặc trưng được thêm vào là các đặc trưng mang tính nhận diện của mẫu dữ liệu. Trong trường hợp các mẫu có sử dụng các biện pháp làm rối như trong **Hình 4** thì việc thêm các best feature giúp nhận ra được các mẫu đó chung một family ransomware so với việc chỉ thêm các vạch section vào là dễ dàng nhận ra hơn.

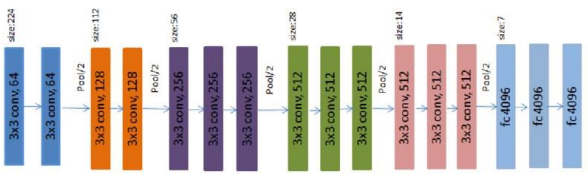


***Hình** 5. Sơ đồ mã hóa các đặc trưng tốt nhất vào hình ảnh*

3.3. Sử dụng Deep learning cho bài toán phân loại ảnh

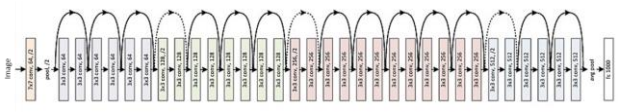
3.3.1. Mô hình

Xây dựng mô hình kết hợp (IMCEC- Image-Based malware classification using ensemble of CNN architectures) giữa VGG16 và ResNet-50 để phân loại các biến thể của loại mã độc Ransom, giữa Ransom với Benign, giữa Ransom với các loại mã độc khác đã được mã hóa thông tin dưới dạng hình ảnh chứa các thông tin PE headers. Sử dụng mô hình Vision Transformer và các mô hình độc lập như ResNet-50 và VGG16 ứng dụng phương pháp Transfer Learning để so sánh với mô hình kết hợp trên.



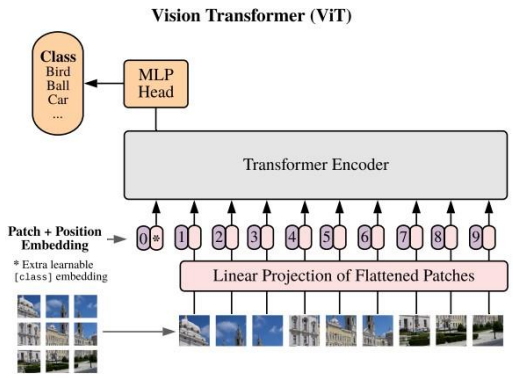
***Hình 6**. Kiến trúc mô hình VGG16*

Mô hình Vgg16 (Simonyan & Zisserman, 2015) gồm 16 lớp mạng (layer) trong đó có 5 khối tích chập (Convolution Layer Block) và 3 lớp Fully Connected Layer dùng để phân loại. Kích thước bộ lọc của mô hình là 3X3, điều này tương tự với lại các layer của các khối (block). Ngoài ra mô hình còn giảm kích thước đầu vào theo các khối nối tiếp nhau, đồng thời tăng số chiều của các Layer. Điều này làm mô hình giảm được tham số (parameters) tính toán khi tăng kích thước và giảm số chiều trong quá trình huấn luyện. Sau mỗi Block sẽ có lớp Pooling với kích thước khoảng trượt (kernel size) là 2x2. Tuy nhiên kích thước của mô hình Vgg16 là rất lớn với hơn 138 triệu parameters, điều này được ResNet cải thiện bằng cách sử dụng kết cấu "nối".



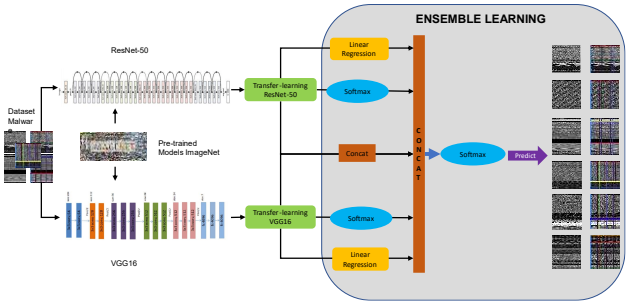
***Hình 7**. Kiến trúc mô hình ResNet-50*

Mô hình ResNet (Krizhevsky, et al., 2012) là mô hình có cấu trúc sử dụng kết nối "tắt" để xuyên qua một hoặc nhiều lớp, điều này giải quyết vấn đề thường hặp phải là Vanishing Gradients nghĩa là tính toán Gradients của các lớp mạng sẽ càng giảm, điều này dẫn đến việc các trong số sẽ không được cập nhiêu trong quá tình huấn luyện. Đặc biệt là những đặc trưng có trọng số gần bằng 0 thì gần như sẽ biến mất trong quá trình huấn luyện. Resnet tương tự như Vgg16 với tích chập là 3X3 nhưng lại thêm tích chập 1X1 bằng nối "tắt" ở khối tương ứng nhánh tiếp theo. Việc thực hiện phép cộng layer trước với layer sau giúp cho mô hình tránh mất mát thông tin, đồng thời tránh trường hợp đạo hàm bằng 0. Điểm đặc biệt của mô hình là các kênh của các khối tích chập tăng dần, điều này sẽ giảm đi kích thước của các block về chiều cao và chiều rộng, giúp mô hình tránh quá phức tạp. Mô hình Resnet có số layer lớn hơn mô Vgg16 với hơn 30 layer, tuy nhiên đối với mỗi block sẽ có lớp Chuẩn hóa hàng loạt (Batch Normalization) để chuẩn hóa các tham số về một khoảng giá trị nhất định thay vì sử dụng tích chập như VGG16.



***Hình 8**. Kiến trúc mô hình Vision Transformer*

Đối với mô hình ResNet và Vgg16, việc sử dụng các lớp mạng tích chập sẽ giúp cho mô hình học tốt hơn. Tuy nhiên trong một số trường hợp, một mô hình không cần phải quá phức tạp, hay không cần phải tập trung vào toàn bộ vị trí của một bức hình thay vào đó chỉ cần tập trung vào những điểm đặc trưng nhất của đối tượng cần phân loại trên hình ảnh. Vision Transformer (Dosovitskiy, et al., 2021)thực hiện điều đó với cơ chế Multihead Attention (Subakan, et al., 2021). Mô hình sẽ chia bức hình thành các Patch và lần lượt đưa từng Patch theo thứ tự vào đầu vào của mô hình, được lấy cảm hứng từ mô hình LSTM và RNN. Các patch sẽ đi qua lớp Flatten để duỗi thẳng bức hình vào thành một vector input. Vì mô hình ViT rất nhạy cảm với vị trí thứ mà LSTM và RNN không cần quan tâm đến (hai mô hình này xử lý theo tuần tự thay vì song song như ViT) nên tương ứng với từng patch sẽ cộng thêm thông tin vị trí vào mỗi vector đầu vào. Qua Multihead Attention, mô hình sẽ tự chú ý vào từng pixel trong bức ảnh và đưa ra trọng số tương ứng, điều này loại bỏ đi những đặc trưng không cần thiết nếu trọng số pixel đó rất thấp xấp xỉ bằng 0.



***Hình 9.** Kiến trúc mô hình IMCEC*

Ensemble learning là phương pháp kết hợp nhiều mô hình với nhau, bổ trợ cho nhau giúp mô hình học tốt hơn. Điều này là phù hợp khi mỗi mô hình có một thế mạnh và điểm yếu riêng, thay vì đổi tham số cho từng mô hình để học tốt hơn, yêu cầu nhiều thời gian nghiên cứu cũng như yêu cầu lượng dữ lớn để huấn luyện để mô hình đạt được đánh giá chuẩn xác nhất. Qua đó chúng tôi đã xây dựng mô hình theo phương pháp Ensemble learning (Vasan, et al., 2020). Bằng cách kết hợp hai mô hình ResNet-50 và VGG16. Điểm mạnh của ResNet là thời gian huấn luyện nhanh, phù hợp với dạng dữ liệu ảnh có kích thước không quá lớn, còn VGG phù hợp với bài toán phân chia đường biên, dữ liệu có kích thước đặc trưng tương đối lớn. Kết hợp hai mô hình sẽ giúp bài toán phân loại trở nên dễ dàng hơn, đạt được độ chính xác cao hơn. Tuy nhiên, nếu kết hợp hai mô hình với số lượng tham số quá lớn sẽ dẫn đến hiện tượng overfitting. Do đó để xây dựng mô hình cần phải có chiến lược huấn luyện hợp lý.

3.3.2. Sử dụng Concatenation lớp layer cuối.

Phương pháp add được áp dụng trong mô hình Resnet, tuy nhiên ở phương pháp concat, được lấy ý tưởng từ mô hình DenseNet (Zhang, et al., 2021) bằng kết nối dày đặc sau đó thực hiện chuẩn hóa bằng một lớp Softmax để chuẩn hóa về kết quả cuối cùng. Nghĩa là tầng cuối cùng sẽ tập hợp tất cả các tầng trước đó thay vì cộng thành một số hạng phức tạp hơn, điều này giúp bảo toàn được đặc trưng của các lớp trước đó với số lượng tham số ít hơn.

3.3.3. Phương pháp Transfer learning

Đối với nhiều bài toán Machine learning, nhiều trường hợp bài toán dự đoán đúng đối với tập test nhưng khi đưa ra thực tế thì kết quả lại rất tệ (Shelhamer, et al., 2016). Nguyên nhân có thể là:

(1) Dữ liệu quá nhỏ không bao quát được tất cả các trường hợp.
(2) Mất cân bằng dữ liệu khi các lớp thuộc nhóm thiểu số lại quá ít so với bộ dữ liệu.
(3) Mô hình quá phức tạp so với số lượng dữ liệu

đưa vào (khoảng vài nghìn hình ảnh) dẫn đến tình trạng bị overfitting.

(4) Qúa trình tối ưu dữ liệu bị khó khăn như thiết lập learning rate chưa tốt, chuẩn hóa kích thước hình ảnh bị mất mát thông tin…

Kỹ thuật Transfer learning giải quyết được bài toán thứ nhất và thứ ba. Bằng cách sử dụng một weights của một pretrain-model đã được huấn luyện trên tập dataset rất lớn. Những mô hình này sẽ được chia làm 2 phần chính. Phần đầu tiên là (mô hình gốc) Base-model tạo ra từ các Conv2D Layer, nhiệm vụ là trích xuất đặc trưng từ input đầu vào của mô hình. Phần thứ hai là lớp Fully Connected Layer (FC) nhiệm vụ thực hiện tính toán phân phối xác xuất để phân loại với quả bằng số lượng lớp muốn phân loại. Đối với mô hình ResNet và VGG16 đã được huấn luyện trên tập ImageNet với hơn 10 triệu ảnh phân loại cho 1000 class. Ta chỉ cần sử dụng lại pretrain-weights của các mô hình này để tiến hành huấn luyện mà không cần phải sử dụng toàn bộ tham số ngay từ đầu. Đối với dữ liệu là mã độc, ta chỉ cần thực hiện Feature extraction, là phương pháp thay đổi cấu trúc của lớp FC cho phù hợp với output mong muốn, bằng cách thay toàn bộ lớp này của mô hình gốc là ResNet và VGG16. Vì dữ liệu mã độc không có trong số lớp phân loại của tập ImageNet nên ta phải thực hiên phá băng (unfrozen) ở layer sâu hơn thì kết quả mới đạt được hiệu quả tốt nhất. Lớp FC được xây dựng bằng cách sử dụng hàm Softmax, Linear Regression, ReLu và hàm chuẩn hóa Normalization thay vì sử dụng lớp tích chập để tránh mô hình phức tạp quá mức.

3.3.4. Chuẩn hóa dữ liệu theo tập ImageNet.

Để sử dụng pretrained-model được xây dựng trên ImageNet, ta phải chuẩn hóa dữ liệu về đúng theo tập ImageNet thì mới đạt được kết quả tối đa. Đối với tập dữ liệu mã độc sau khi đã mã hóa hình ảnh, giá trị pixel của ảnh sẽ có giá trị từ 0 đến 255. Việc giá trị lớn như vậy sẽ ảnh hưởng đến kết quả của bài toán nên phải đưa về dãy [0,1] sau đó chuẩn hóa phù hợp với yêu cầu hay cụ thể hơn là theo chuẩn ImageNet. Trước khi huấn luyện phải thực hiện resize về kích thước cố định là $224x224$, khi đó dùng hàm chuẩn hóa các giá trị pixel theo giá trị trung bình và độ lệch chuẩn của ImageNet là $mean = [0.485, 0.456, 0.406]$ và $std = [0.229, 0.224, 0.225]$ và output của pixel sẽ được tính theo công thức.

$$\text{output[n]} = (\text{input[n]} - \text{mean[n]}) / \text{std[n]}$$

n: số kênh của bức ảnh đầu vào, n=3 nếu là ảnh RGB.

$mean$: giá trị trung bình

std: độ lệch chuẩn (standard deviation)

3.3.5. Kỹ thuật cân bằng trọng số - balanced class weights.

Với bài toán giải quyết phía trên, việc mất cân bằng dữ liệu khi nhóm thiểu số là quá thấp so với nhóm đa số. Điều này dẫn đến bài toán sẽ bị lệch hẳn sang bên nhóm đa số (bias). Để giải quyết vấn đề này ta thực hiện thêm balance class weights sau mỗi bước dự đoán output của mô hình. Balance class weights này sẽ được tính theo công thức:

$$w_j = \text{n\_samples} / (\text{n\_classes} * \text{n\_samples}_j)$$

w_j: weight của lớp thứ j

$n\_samples$: số lượng toàn bộ dữ liệu để huấn luyện.

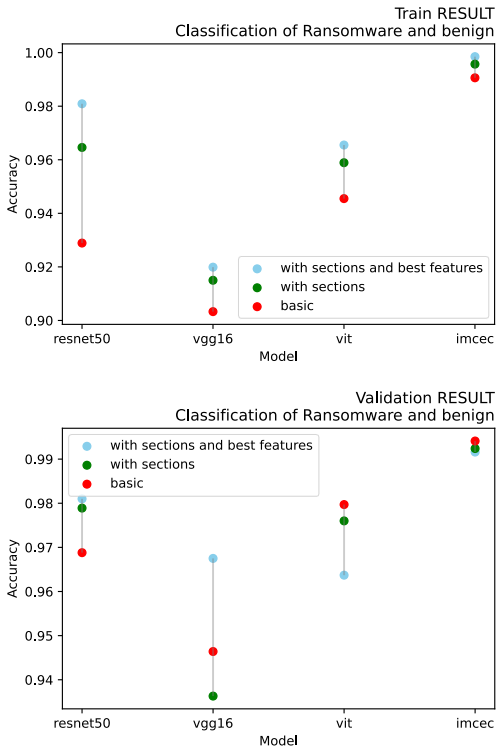
$n\_classes$: số lượng lớp cần phân loại.

n_samples_j: là số lượng mẫu của lớp thứ j.

Với trọng số w sẽ được nhân với kết quả dự đoán khi tính toán hàm loss trong quá trình huấn luyện của mô hình. Nếu dữ liệu ở nhóm thiểu số thì w_j của nhóm đó sẽ lớn và ngược lại đối với nhóm đa số.

4. Thực nghiệm và đánh giá kết quả.

4.1. Phân loại Ransomware với Benign

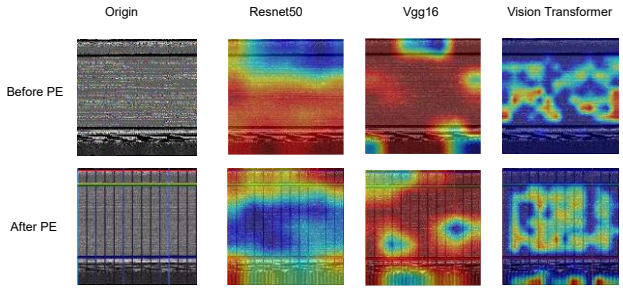


Hình 10. *Kết quả khi chạy mô hình trên tập dữ liệu để phân biệt ransomware và benign*

Ở bài toán đầu tiên, chúng tôi tiến hành thử nghiệm và đánh giá giữa việc sử dụng section và kết hợp sections với các PE headers feature để so sánh với dạng ảnh thông thường để so sánh mức độ ảnh hưởng của việc import các chuỗi feature lên image.

Kết quả cho thấy việc thêm các chuỗi section đã thực sự làm nổi bật khả năng phân loại hình ảnh. Thêm vào đó, việc kết hợp headers feature đã càng làm tăng thêm khả năng phân loại giữa ransomeware và benign so với các dạng ảnh Gray thông thường. Đối với mô hình Vision Transformer, kết quả của ba thử nghiệm có độ chính xác không chênh lệch nhau quá nhiều, nghĩa là việc thêm PE headers và section không thật sự cải thiện chất lượng của mô hình. Nguyên nhân của việc này bởi vì đầu vào của mô hình phụ thuộc vào kích thước của Patch, nên khi thêm các đặc trưng như PE theo chiều dọc và chiều ngang có thể bị mất đi thông tin khi bị cắt rời ra trong quá trình huấn luyện do đó ảnh hưởng đến việc mất mát thông tin PE. Mô hình IMCEC, với độ chính xác lên đến hơn 99% có thể thấy sức mạnh của Ensemble learning khi kết hợp các mô hình lại với nhau. Vì kích thước của mô hình rất lớn nên việc thêm PE không ảnh hưởng quá nhiều đến chất lượng kết quả đầu ra. Đặc trưng của các thuộc nhóm thiểu số trong khi mô hình lại phân tích toàn bộ đặc trưng của cả bức hình, do đó kết quả gần như tương đương

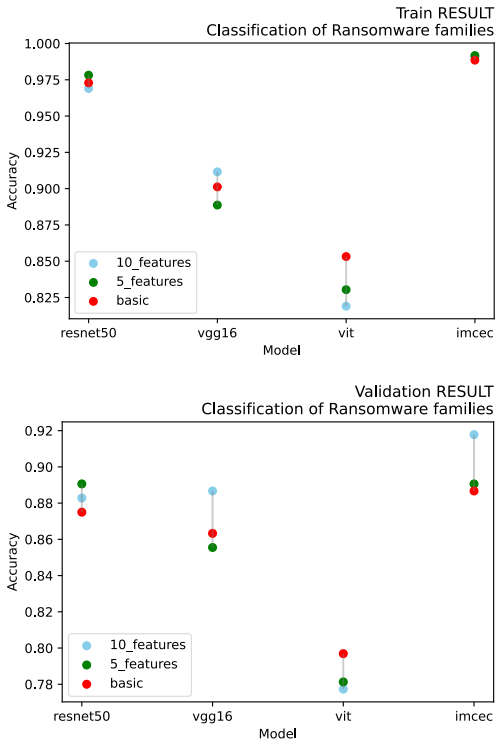
nhau. Thông qua thực nghiệm ban đầu, chúng tôi nhận định được tầm quan trọng của việc áp dụng các header features và sections lên ảnh mã độc. Vấn đề tiếp theo của chúng tôi là sẽ áp dụng bao nhiêu header features để đạt được kết quả tối ưu nhất.



Hình 11. *Kết quả đánh giá bằng phương pháp GradCam*

Để đánh giá mô hình trên tập dữ liệu sau khi huấn luyện, chúng tôi dùng phương pháp Grad Cam (Selvaraju, et al., 2016) để phân tích mức độ tập trung của mô hình vào các vị trí trên bức ảnh chứa ransomeware. Ở **Hình 11**, Vị trí càng đậm (màu đỏ) là nơi mô hình tập trung vào đối tượng để phân loại. Mô hình VGG16 gần như tập trung vào toàn bộ vị trí của bức hình nên kết quả khi chứa PE headers, section và ngược lại không có những thông tin này thì có kết quả chênh lệch nhiều. Mô hình Resnet khi thêm các đặc trưng PE ảnh hưởng tương đối lớn đến kết quả phân loại khi mô hình phân chia ra bởi khu vực chứa những thông tin quan trọng. Mô hình ViT có kết quả gần giống nhau khi mô hình vẫn chỉ tập trung vào một số khu vực nhất định và việc thêm PE thực sự chưa ảnh hưởng đến kết quả cuối cùng.

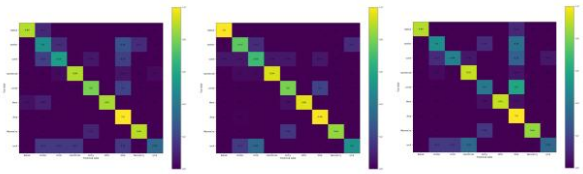
4.2. Phân loại các biến thể của ransomware.



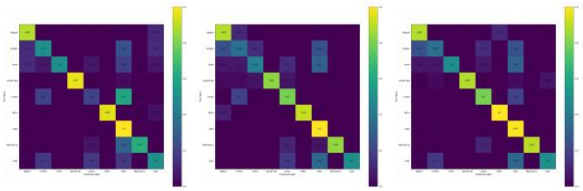
Hình 12. *Kết quả khi chạy mô hình trên tập dữ liệu để phân biệt các ransomware families*

Chuẩn bị một dataset gồm các mẫu ransomware đến từ các families đã được phân loại kĩ lưỡng, từ đó thêm vào hình ảnh đại diện mẫu 5 đặc trưng và 10 đặc trưng.

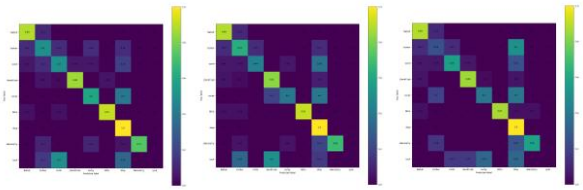
Kết quả thể hiện tại **Hình 12** đã có thể kết luận được rằng đối với số lượng các đặc trưng sẽ là một siêu tham số (Hyperparameter) phải thay đổi trong quá trình sử dụng phương pháp này. Thêm bao nhiêu đặc trưng của PE header sẽ ảnh hưởng đến kết quả cuối cùng, không phải thêm càng nhiều thì đạt kết quả càng cao, mô hình chỉ đạt được trạng thái tốt nhất khi thêm đặc trưng PE header phù hợp với mô hình và tập dữ liệu. Trong đó việc thêm 10 đặc trưng ở trong thử nghiệm này tỏ là một thông số phù hợp với mô hình VGG16 và IMCEC. Mô hình ViT thì không phân loại tốt khi thêm PE bởi cơ chế hoạt động của mô hình này không phù hợp với cách thêm các đặc trưng dưới dạng các vạch kẻ mã hóa màu.



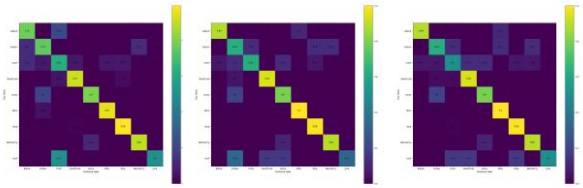
Hình 13*. Confusion Matrix của mô hình ResNet-50*



Hình 14. *Confusion Matrix của mô hình VGG16*



Hình 15. *Confusion Matrix của mô hình Vision Transformer*



Hình 16*. Confusion Matrix của mô hình IMCEC*

4.3. Phân loại Ransomware với các loại mã độc khác.

Thêm những đặc trưng về PE ảnh hưởng phần nào đến kết quả của các mô hình, tuy nhiên việc thay đổi này không ảnh hưởng quá lớn đến chất lượng của mô hình. Đặc biệt là mô hình VGG16 dường như việc không thêm thông tin PE cho ra kết quả tốt hơn. Trong khi đó mô hình ResNet và IMCEC đạt được độ chính xác cao hơn trong trường hợp trích xuất PE theo phương pháp hiện tại.

Bảng 3.
Binary classification between ransomware and other malware (%).

	Dataset					
	Sections and 10 best features		Sections and 5 best features		Gray scale	
	Train	Val	Train	Val	Train	Val
Resnet50	96.444	96.88	97.18	96.29	94.83	96.48
VGG16	92.09	94.14	89.81	92.77	93.20	95.51
ViT	95.41	95.70	93.00	93.81	96.00	95.31
IMCEC	99.65	98.44	99.26	97.85	99.10	97.27

Val = Validation

5. Kết luận

Hình ảnh thang xám là một phương pháp phổ biến được sử dụng rộng rãi trong các cách thức phân tích tĩnh mã độc dựa vào hình ảnh và deep learning. Trong nghiên cứu này, chúng tôi đã đề xuất một phương pháp tích hợp thêm các thông tin có trong PE headers mà trong hình ảnh thang xám thông thường sẽ không thể cung cấp được cho mô hình. Bên cạnh đó, chúng tôi còn đề xuất cách để mã hóa các thông tin này vào trong hình ảnh. Từ đó, chúng tôi xây dựng một mô hình kết hợp giữa VGG16 và ResNet50 và thử nghiệm với mô hình sử dụng cơ chế Attention với bài toán này. Kết quả trên các thử nghiệm cho thấy, việc thêm các thông tin đặc trưng của PE header vào có thể giúp nâng cao độ chính xác của tác vụ phân biệt và phát hiện mã độc tống tiền.

Bởi vì các mã độc sẽ luôn được viết lại và làm nhiễu hoặc che dấu thông tin trong PE header sau khi nén, thì phương pháp này sẽ không thực sự là thể hiện được đặc trưng của mã độc đó.

Tài liệu tham khảo

Blog Google AI, 2019. *Turbo, An Improved Rainbow Colormap for Visualization.*, s.l.: [online] Available at: https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html [Accessed 8 Feb. 2022]..

Daniel, G., Carles, M. & Jordi, P., 2020. *The rise of machine learning for detection and classification of malware: Research developments, trends and challenges,* s.l.: Volume 153, 102526, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2019.102526..

Dosovitskiy, A. et al., 2021. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,* s.l.: https://arxiv.org/abs/2010.11929.

Kalita, A. B., Abudawaood, N. & Jugal, 2020. *Classifying Malware Images with Convolutional Neural Network Models,* s.l.: DOI: 10.6633/IJNS.202011_22(6).17.

Kancherla, K. & Mukkamala, S., 2013. *Image visualization based malware detection,* s.l.: DOI: 10.1109/CICYBS.2013.6597204.

Krizhevsky, A., Sutskever, I. & Hinton, G., 2012. *ImageNet Classification with Deep Convolutional Neural Networks,* s.l.: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Nataraj, L., Karthikeyan, S., Jacob, G. & Manjunath, B. S., 2011. *Malware Images: Visualization and Automatic Classification,* s.l.: Proceedings of the 8th International Symposium on Visualization for Cyber Security. https://doi.org/10.1145/2016904.2016908.

Rezende, E. et al., 2017. *Malicious Software Classification Using Transfer Learning of ResNet-50 Deep Neural Network,* s.l.: doi: 10.1109/ICMLA.2017.00-19.

Selvaraju, R. R. et al., 2016. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,* s.l.: https://arxiv.org/abs/1610.02391.

Shelhamer, E., Long, J. & Darrell, T., 2016. *Fully Convolutional Networks for Semantic Segmentation,* s.l.: doi: 10.1109/TPAMI.2016.2572683.

Simonyan, K. & Zisserman, A., 2015. *Very Deep Convolutional Networks for Large-Scale Image Recognition,* s.l.: https://arxiv.org/abs/1409.1556.

Subakan, C. et al., 2021. *Attention Is All You Need In Speech Separation,* s.l.: doi: 10.1109/ICASSP39728.2021.9413901.

Vasan, D. et al., 2020. *Image-Based malware classification using ensemble of CNN architectures (IMCEC),* s.l.: doi: https://doi.org/10.1016/j.cose.2020.101748.

Xiao, M. et al., 2021. *Image-based malware classification using section distribution information,* s.l.: Computers & Security, [online] 110, p.102420. Available at: https://www.sciencedirect.com/science/article/pii/S0167404821002443 [Accessed 8 Feb. 2022]..

Zhang, C. et al., 2021. *ResNet or DenseNet? Introducing Dense Shortcuts to ResNet,* s.l.: https://arxiv.org/abs/2010.12496.