**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN**

---------



**BÁO CÁO ĐỒ ÁN**

**TÌM HIỂU THUẬT TOÁN CNN CHO PHÂN LOẠI MÃ ĐỘC PE**

**Sinh viên thực hiện:**

Trần Thanh Sơn - 21522557

Trần Trọng Nhân - 21522411

**Giảng viên:**

TS. Nguyễn Tấn Cầm

Thành phố Hồ Chí Minh, tháng 12 năm 2023

**Abstract**—Vấn đề phát hiện và phân loại mã độc đang dần trở nên khó khăn hơn khi mã độc đang phát triển mạnh mẽ và có sức lây lan rộng trên quy mô toàn cầu. Nhu cầu phát hiện và phân loại mã độc tăng mạnh kéo theo sự ra đời của nhiều công nghệ và phương pháp mới hỗ trợ đáp ứng nhu cầu này. Một trong những phương pháp đang được các nhà nghiên cứu quan tâm đó là phương pháp deep learning với sự phát triển của nhiều mô hình khác nhau qua các thế hệ. Ở bài nghiên cứu này, chúng tôi đề xuất thử nghiệm các mô hình CNN tiêu biểu cho việc phân loại mã độc PE. Thông qua các mô hình này, chúng tôi so sánh hiệu suất và chọn ra mô hình tốt nhất với độ chính xác cao.

**Keywords**—Malware classification, convolutional neural network (CNN), portable executable (PE), deep learning

# GIỚI THIỆU

Mã độc (malicious software hay malware) là chương trình máy tính độc hại được tạo ra với mục đích xấu nhằm gây hậu quả tiêu cực cho hệ thống, dữ liệu, người sử dụng máy tính hoặc giành quyền truy cập trái phép vào những thông tin nhạy cảm [1]. Ở bài nghiên cứu này, phạm vi của malware được giới hạn ở định dạng PE (Portable Executable). PE là định dạng file thực thi chủ yếu trên Win32 bao gồm các đuôi file như: EXE, DLL (32 bit), NET, COM, CPL,… trừ định dạng VxDs và DLL (16 bit) [2]. Theo thống kê của AV-TEST, năm 2021 ghi nhận tổng cộng gần 120 triệu mã độc Windows [3]. Điều này cho thấy mã độc Windows có tốc độ phát triển nhanh chóng, nếu không được phát hiện và ngăn chặn kịp thời thì sẽ gây ra hậu quả khó lường đến các hệ thống thông tin trong tương lai.

Phương pháp học máy (machine learning) và học sâu (deep learning) giữ vai trò không thể thiếu trong việc phát hiện và phân loại mã độc PE nói riêng và tất cả mã độc nói chung. Kỹ thuật học máy cộng với việc phân tích PE Header sẽ giúp việc phát hiện và phân loại mã độc PE trở nên dễ dàng và hiệu quả [4].

Một vài nghiên cứu gần đây đã sử dụng một mô hình deep learning tiên tiến để phân loại mã độc có tên gọi là mạng nơ-ron tích chập (CNN). CNN hỗ trợ việc phân loại mã độc thông qua hình ảnh dựa trên các đặc trưng được trích xuất nhờ việc được huấn luyện theo tập dữ liệu cụ thể [5].

Đóng góp chính của chúng tôi trong nghiên cứu này là sử dụng các mô hình huấn luyện trước (pre-trained) từ thư viện Keras cho việc phân loại mã độc PE, huấn luyện các mô hình này trên nền tảng Kaggle với các bộ dữ liệu, so sánh hiệu suất giữa các mô hình và chọn ra mô hình hiệu quả nhất.

Bài nghiên cứu này được bố cục thành các chương như sau: Chương 2 trình bày các công trình liên quan nghiên cứu việc phân loại mã độc. Chương 3 trình bày các bộ dữ liệu được sử dụng để huấn luyện các mô hình CNN. Chương 4 trình bày phương pháp đề xuất về việc thử nghiệm các mô hình CNN trên các bộ dữ liệu. Chương 5 trình bày kết quả thực nghiệm của các mô hình CNN trên các bộ dữ liệu, đồng thời so sánh các bộ dữ liệu. Chương 6 là kết luận.

# CÔNG TRÌNH LIÊN QUAN

Phần này giới thiệu các hướng tiếp cận giải quyết vấn đề phân loại mã độc.

Bài [6] đề xuất sử dụng kết hợp ba phương pháp khác nhau là TF-IDF, Paragraph Vector và API call sequences để phân loại mã độc với lợi thế của Xử lý ngôn ngữ tự nhiên (NLP).

Bài [7] đề xuất một mô hình học ngôn ngữ của mã độc thông qua các chỉ thị được thực thi và trích xuất các đặc trưng mạnh mẽ trong miền thời gian. Bộ phân loại tiêu chuẩn sử dụng các đặc trưng này để phát hiện và phân loại mã độc.

Bài [8] đề xuất XGB với kỹ thuật Loại bỏ tính năng ngược dựa trên phiếu bầu (XGB-VBFE) để chọn các đặc trưng tối ưu nhằm xây dựng mô hình và phân loại các mã độc với độ chính xác cao hơn.

Bài [9] đề xuất một phương pháp mã hóa và sắp xếp byte từ tệp nhị phân thành hình ảnh cho việc phân loại mã độc.

Bài [10] đề xuất một mô hình nhận dạng mã độc dựa trên mạng nơ-ron đồ thị (GNN), chuyển đổi hành vi mã độc thành cấu trúc dữ liệu biểu đồ. Các biểu đồ được tạo sẽ được đưa vào mạng GNN để mạng học cấu trúc mã nhằm xác định các lớp mã độc.

Bài [11] đề xuất một framework dựa trên CNN mang lại hiệu suất tốt hơn so với các phương pháp truyền thống của học nông (shallow learning) cho việc phân loại mã độc sử dụng các hình ảnh Byteplot.

Bài [12] đề xuất sử dụng, so sánh các kỹ thuật phân cụm (clustering) và xem xét tác động của việc tăng kích thước và số lượng cụm đến độ chính xác của phân cụm. Bằng việc tính toán các cụm bằng thuật toán K-means và thuật toán cực đại hóa kỳ vọng (EM), với điểm số cơ bản dựa trên mô hình Markov ẩn (HMM), clustering có thể đạt đến độ chính xác cao trong việc tự động phân loại mã độc mà không cần huấn luyện lại mô hình.

Bài [13] đề xuất một kiến trúc phân loại mã độc sử dụng mạng nơ-ron đa nhiệm (MtNet). Kiến trúc này mang lại những cải tiến nhỏ cho deep learning với ngưỡng phát hiện là 0.5. Một tệp được dự đoán là malware nếu xác suất độc hại (malicious) vượt quá xác suất lành tính (benign).

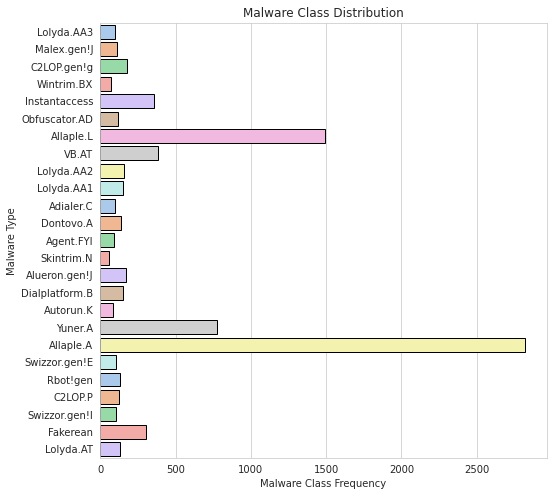
Bài [14] đề xuất HYDRA, một framework đa phương thức để giải quyết bài toán phát hiện và phân loại mã độc bằng cách kết hợp nhiều loại đặc trưng khác nhau để tìm ra mối quan hệ giữa các phương thức riêng biệt. Framework này học từ nhiều nguồn thông tin khác nhau để tối đa hóa lợi ích của nhiều loại đặc trưng để phản ánh các đặc điểm của các mã độc. Một hệ thống đường cơ sở bao gồm các thành phần được thiết kế thủ công và end-to-end để kết hợp lợi ích của kỹ thuật đặc trưng và deep learning nhằm biểu diễn các đặc điểm của mã độc một cách hiệu quả.

Bài [15] đề xuất phương pháp phân loại mã độc bằng thuật toán deep learning dựa trên thông tin byte. Phương pháp đề xuất sử dụng hình ảnh được tạo từ thông tin byte của mã độc có thể phản ánh bối cảnh hành vi của mã độc, và sentence analysis dựa trên CNN được sử dụng để xử lý các hình ảnh được tạo.

# BỘ DỮ LIỆU

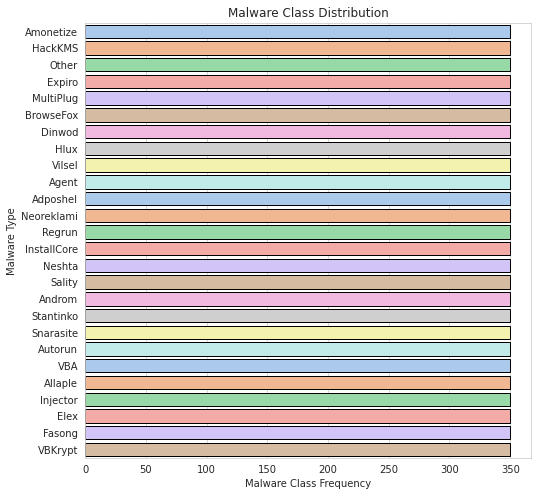
Các mô hình CNN được huấn luyện và đánh giá dựa trên ba bộ dữ liệu khác nhau: Bộ dữ liệu Malimg, bộ dữ liệu Malevis và bộ dữ liệu Blended. Bộ dữ liệu Blended được tạo ra bằng cách kết hợp 5 lớp mã độc chính từ bộ dữ liệu Malimg vào 25 lớp mã độc của bộ dữ liệu Malevis. Từ đó, bộ dữ liệu Malimg được xem như là một bộ dữ liệu mất cân bằng hoàn toàn, bộ dữ liệu Blended như một bộ dữ liệu mất cân bằng vừa phải và bộ dữ liệu Malevis như một bộ dữ liệu cân bằng hoàn toàn.

Bộ dữ liệu Malimg chứa 9339 hình ảnh thước xám (grayscale) của các mã độc từ 25 lớp mã độc riêng biệt [16]. Hình 1 mô tả sự phân bố các mẫu mã độc của Malimg.



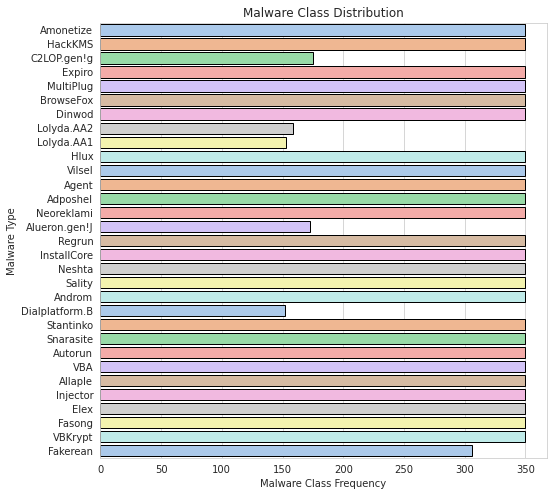
Hình 1. Mô tả bộ dữ liệu Malimg

Bộ dữ liệu Malevis cung cấp bộ dữ liệu thực tế dựa trên RGB để đánh giá các nghiên cứu nhận dạng mã độc đa lớp dựa trên thị giác. Bộ dữ liệu Malevis bao gồm tổng cộng 9100 hình ảnh RGB cho quá trình huấn luyện và 5126 hình ảnh RGB cho quá trình kiểm thử. Tất cả các lớp huấn luyện đều có 350 mẫu hình ảnh trong khi bộ kiểm thử có số lượng hình ảnh khác nhau [17]. Hình 2 mô tả sự phân bố các mẫu mã độc của Malevis.



Hình 2. Mô tả bộ dữ liệu Malevis

Bộ dữ liệu Blended dùng cho các mô hình CNN xử lý đồng thời các hình ảnh RGB và grayscale. Điều này nhằm mục đích là để phân loại các hình ảnh mã độc đa lớp [18]. Hình 3 mô tả sự phân bố các mẫu mã độc của Blended.



Hình 3. Mô tả bộ dữ liệu Blended

# PHƯƠNG PHÁP ĐỀ XUẤT

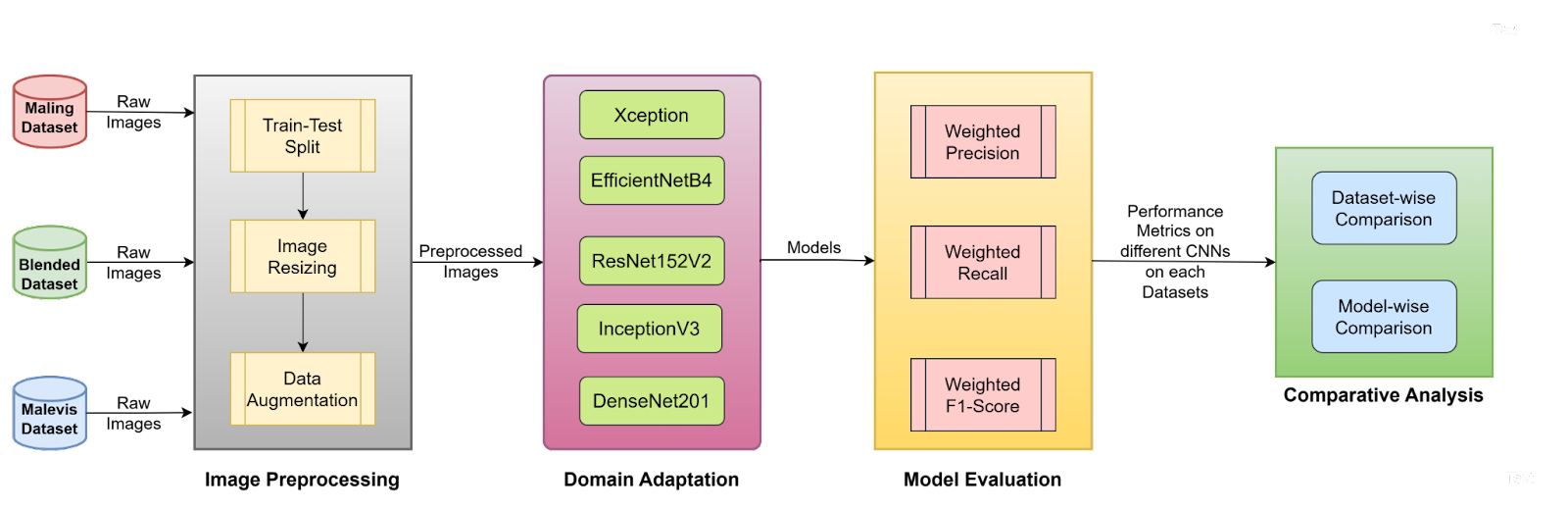
Trong bài báo này, chúng tôi đề xuất thử nghiệm năm mô hình CNN cho việc phân loại mã độc PE. Ở phần này, chúng tôi mô tả quy trình thực hiện phân loại mã độc PE với năm mô hình CNN. Các mô hình được huấn luyện trên nền tảng Kaggle với bộ gia tốc GPU P100.

## Kiến trúc

Hình 4 thể hiện sơ đồ kiến trúc cho luồng dữ liệu để so sánh sự phân loại ảnh mất cân bằng của ba bộ dữ liệu mã độc khác nhau. Có hai bộ dữ liệu hình ảnh mã độc là Malimg và Malevis, được kết hợp thành một bộ dữ liệu duy nhất có sự mất cân bằng vừa phải giữa các lớp.

Tất cả các hình ảnh từ các bộ dữ liệu tương ứng đều trải qua giai đoạn tiền xử lý ảnh bao gồm việc thay đổi kích thước và tăng cường. Ở cuối giai đoạn này, mỗi bộ dữ liệu có được tập huấn luyện, tập xác thực và tập thử nghiệm. Tiếp theo, năm mô hình CNN được thử nghiệm trên mỗi bộ dữ liệu và được đánh giá dựa trên các chỉ số là Precision, Recall và F1-score.

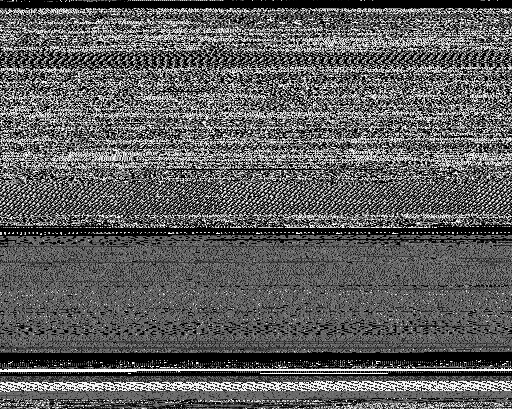
Các chỉ số của từng mô hình được sử dụng để đánh giá và so sánh sự biến đổi hiệu suất mô hình dựa trên sự mất cân bằng giữa các lớp mã độc.



Hình 4. Sơ đồ kiến trúc

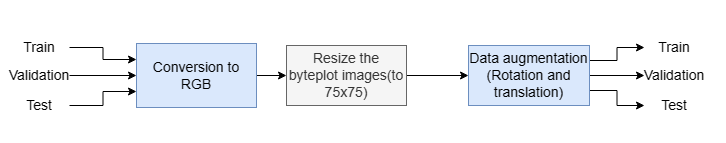
## Tiền xử lý ảnh

Tiền xử lý ảnh là một bước tiền đề để chuẩn hóa và tăng cường dữ liệu hình ảnh trước khi đưa vào một mạng nơ-ron để huấn luyện. Trong số các bộ dữ liệu, bộ dữ liệu Malimg bao gồm các hình ảnh grayscale, bộ dữ liệu Malevis bao gồm các hình ảnh RGB và bộ dữ liệu Blended bao gồm cả hình ảnh grayscale và hình ảnh RGB được thể hiện ở hình 5. Vì bộ dữ liệu Malimg có các hình ảnh với kích thước khác nhau nên cần chuyển đổi kích thước của các hình ảnh đó thành một kích thước hình ảnh duy nhất.

Hình 5. Hình ảnh grayscale (bên trái) và hình ảnh RGB (bên phải)

Sau khi dữ liệu được chia thành tập huấn luyện, tập xác thực và tập thử nghiệm, tất cả hình ảnh được chuyển đổi sang RGB và thay đổi kích thước thành 75 pixel với chiều rộng và chiều cao bằng nhau. Sau đó, các hình ảnh được tăng cường bằng cách xoay và dịch chuyển ảnh làm cho mô hình bất biến. Hình 6 thể hiện sơ đồ quy trình tiền xử lý ảnh.

****

Hình 6. Quy trình tiền xử lý ảnh

## Domain adaptation

Domain adaptation trong học chuyển giao (transfer learning) đề cập đến quá trình sử dụng kiến thức thu được từ một miền (domain) để cải thiện hiệu suất mô hình trong một miền khác liên quan. Điều này bao gồm việc chuyển giao kiến thức từ một miền nguồn sang một miền đích, nơi phân bố dữ liệu có thể khác nhau.

Sự phân bố dữ liệu của các hình ảnh byteplot sẽ được học bởi mô hình sau khi làm một số lớp trích xuất đặc trưng có thể được huấn luyện. Ngoài ra, các lớp dropout regularization được thêm vào kiến trúc để giảm khả năng overfitting đối với bộ dữ liệu được cân bằng hoàn toàn. Thêm vào đó, các mô hình được huấn luyện với early stopping bằng cách theo dõi validation loss của mỗi epoch với cùng mục tiêu là tránh bị overfitting.

Các mô hình CNN được chọn để thử nghiệm trên mỗi bộ dữ liệu là XceptionNet, EfficientNetB4, ResNet152V2, InceptionV3 và DenseNet201.

## Thước đo đánh giá mô hình

Các chỉ số đánh giá được sử dụng để đánh giá hiệu suất của một mô hình CNN thông qua mức độ phù hợp của các dự đoán mà mô hình đưa ra với nhãn thực tế của dữ liệu được sử dụng để huấn luyện và kiểm thử mô hình. Trong bài toán phân loại đa lớp không cân bằng, chỉ số chung như accuracy không thích hợp cho việc đánh giá mô hình vì nó không xem xét số lượng ảnh hỗ trợ có sẵn cho mỗi lớp. Do đó, các chỉ số phù hợp hơn như precision, recall và F1-score được tính dựa trên trọng số là lựa chọn thích hợp nhất.

Trong dữ liệu kiểm thử, mỗi lớp có số lượng ảnh cụ thể để đánh giá (ký hiệu là wi), precision (ký hiệu là pi), recall (ký hiệu là ri), và F1-score (ký hiệu là f1i) được tính toán so với thực tế.

Weighted Precision =

Weighted Recall =

Weighted F1-score =

Các chỉ số weighted được sử dụng để chuẩn hóa hiệu suất của các mô hình trên các bộ dữ liệu có độ mất cân bằng khác nhau. Trong phần sau, các mô hình sẽ được so sánh bằng cách sử dụng các chỉ số này.

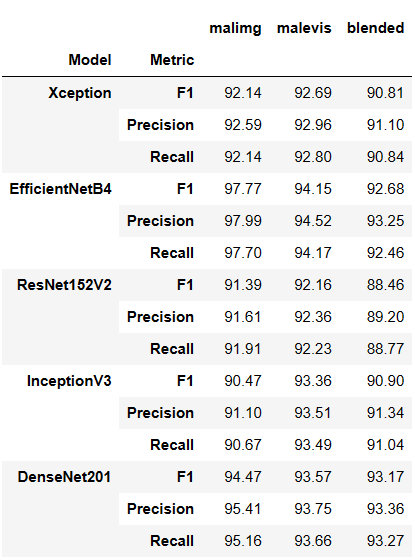
# KẾT QUẢ THỰC NGHIỆM

Phần này chúng tôi so sánh và đánh giá hiệu suất của các mô hình CNN trên từng bộ dữ liệu, đồng thời chúng tôi so sánh các bộ dữ liệu.

## So sánh giữa các mô hình

Trong lĩnh vực học máy, các bộ phát hiện mã độc thường là các bộ phân loại đa lớp. Ở phần này, chúng tôi so sánh khả năng phân loại hình ảnh mã độc đa lớp của năm mô hình CNN. Bảng 1 cho thấy các chỉ số đánh giá của năm mô hình CNN trên ba bộ dữ liệu. Từ các kết quả có thể nhận thấy rằng bộ dữ liệu càng cân bằng thì sự khác biệt về hiệu suất của các mô hình càng nhỏ.

Bảng 1. So sánh các chỉ số đánh giá giữa các mô hình CNN (%)



Có sự khác biệt không nhỏ về các chỉ số đánh giá của năm mô hình trên bộ dữ liệu Malimg. Mô hình EfficientNetB4 có các chỉ số cao nhất bao gồm precision, recall và F1-score lần lượt là 97.99%, 97.70% và 97.77%. Đối với bộ dữ liệu Blended, mô hình DenseNet201 đạt hiệu suất tốt nhất trong các mô hình được thử nghiệm với precision, recall và F1-score lên tới khoảng 93%. Đối với bộ dữ liệu Malevis, hầu hết các mô hình đều hoạt động tốt với tỉ lệ trên khoảng 92% do sự phân bố đồng đều giữa các lớp mã độc.

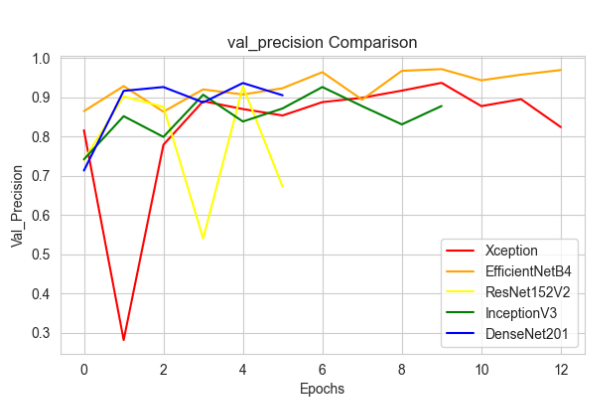
Bảng 2 so sánh chỉ số precision trung bình giữa các mô hình trên ba bộ dữ liệu. Có thể thấy, mô hình EfficientNetB4 dẫn đầu với tỉ lệ cao nhất là 95.25%.

Bảng 2. So sánh precision trung bình trên ba bộ dữ liệu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | XceptionNet | EfficientNetB4 | ResNet152V2 | InceptionV3 | DenseNet201 |
| Average precision (%) | 92.22 | 95.25 | 91.06 | 91.98 | 94.17 |

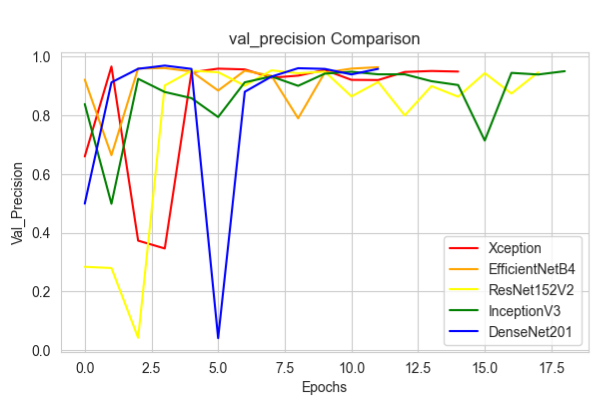
Precision là chỉ số quan trọng và tin cậy nhất khi đánh giá một bộ phân loại mã độc vì đại lượng false positive có giá trị hơn đại lượng false negative. Do đó, sự thay đổi validation precision của các mô hình CNN được xem xét dựa theo các epoch trên từng bộ dữ liệu.

Hình 7 biểu diễn biểu đồ đường thể hiện validation precision của từng mô hình CNN trên bộ dữ liệu Malimg.



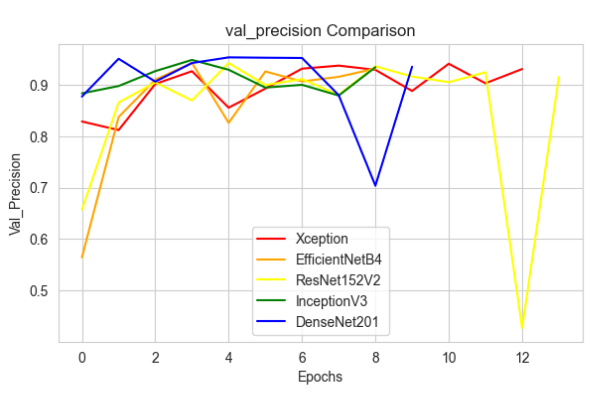
Hình 7. Validation precision trên bộ dữ liệu Malimg

Hình 8 biểu diễn biểu đồ đường thể hiện validation precision của từng mô hình CNN trên bộ dữ liệu Malevis.



Hình 8. Validation precision trên bộ dữ liệu Malevis

Hình 9 biểu diễn biểu đồ đường thể hiện validation precision của từng mô hình CNN trên bộ dữ liệu Blended.

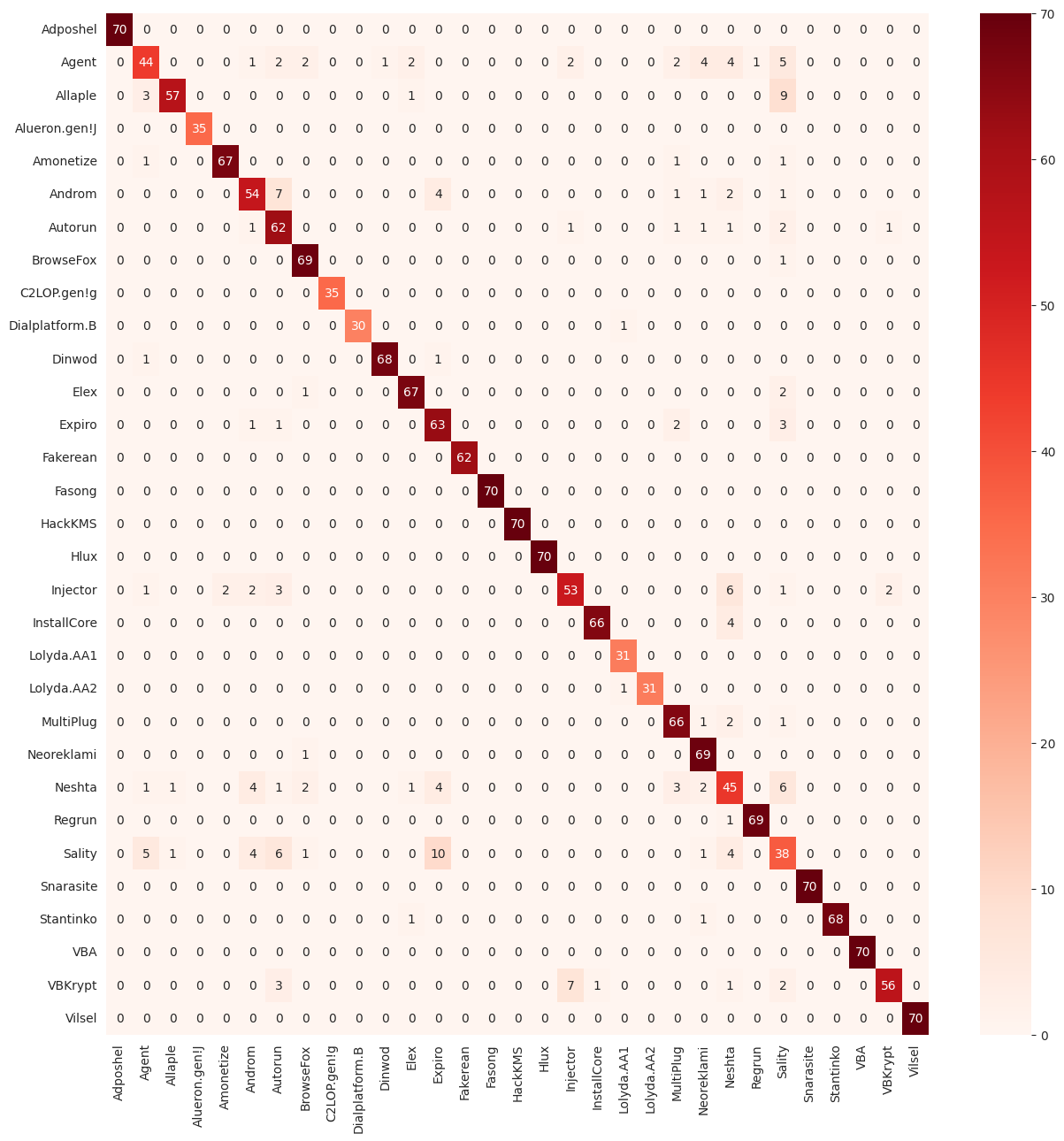


Hình 9. Validation precision trên bộ dữ liệu Blended

Các mô hình EfficientNetB4, InceptionV3 và DenseNet201 đều hoạt động tốt với cả dữ liệu cân bằng và mất cân bằng. Mô hình XceptionNet hoạt động tốt trong mỗi epoch đối với bộ dữ liệu Blended và bộ dữ liệu Malevis nhưng không thể học tốt từ bộ dữ liệu mất cân bằng hoàn toàn như Malimg. Mô hình ResNet152V2 không học được từ dữ liệu mất cân bằng nhưng hoạt động tốt đối với dữ liệu cân bằng.

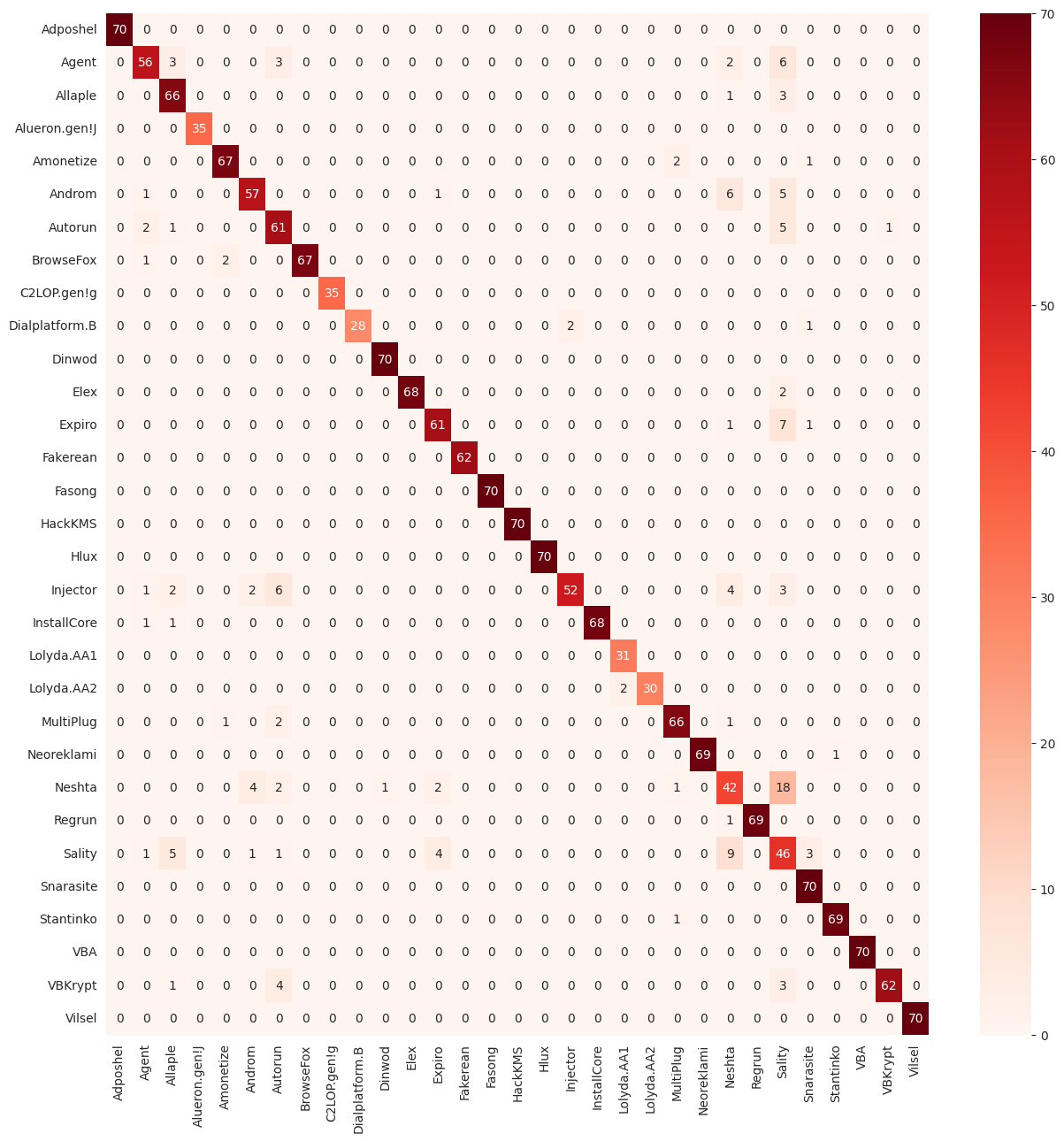
Bộ dữ liệu Blended là bộ dữ liệu mất cân bằng được kết hợp giữa bộ dữ liệu Malimg và bộ dữ liệu Malevis nên chúng tôi khảo sát từng mô hình thông qua confusion matrix với bộ dữ liệu này. Các phần tử trên đường chéo thể hiện số lượng các mẫu hình ảnh mã độc PE được phân loại đúng. Các phần tử ngoài đường chéo thể hiện số lượng các mẫu hình ảnh mã độc PE được phân loại sai.

Hình 10 thể hiện confusion matrix của mô hình XceptionNet với bộ dữ liệu Blended.



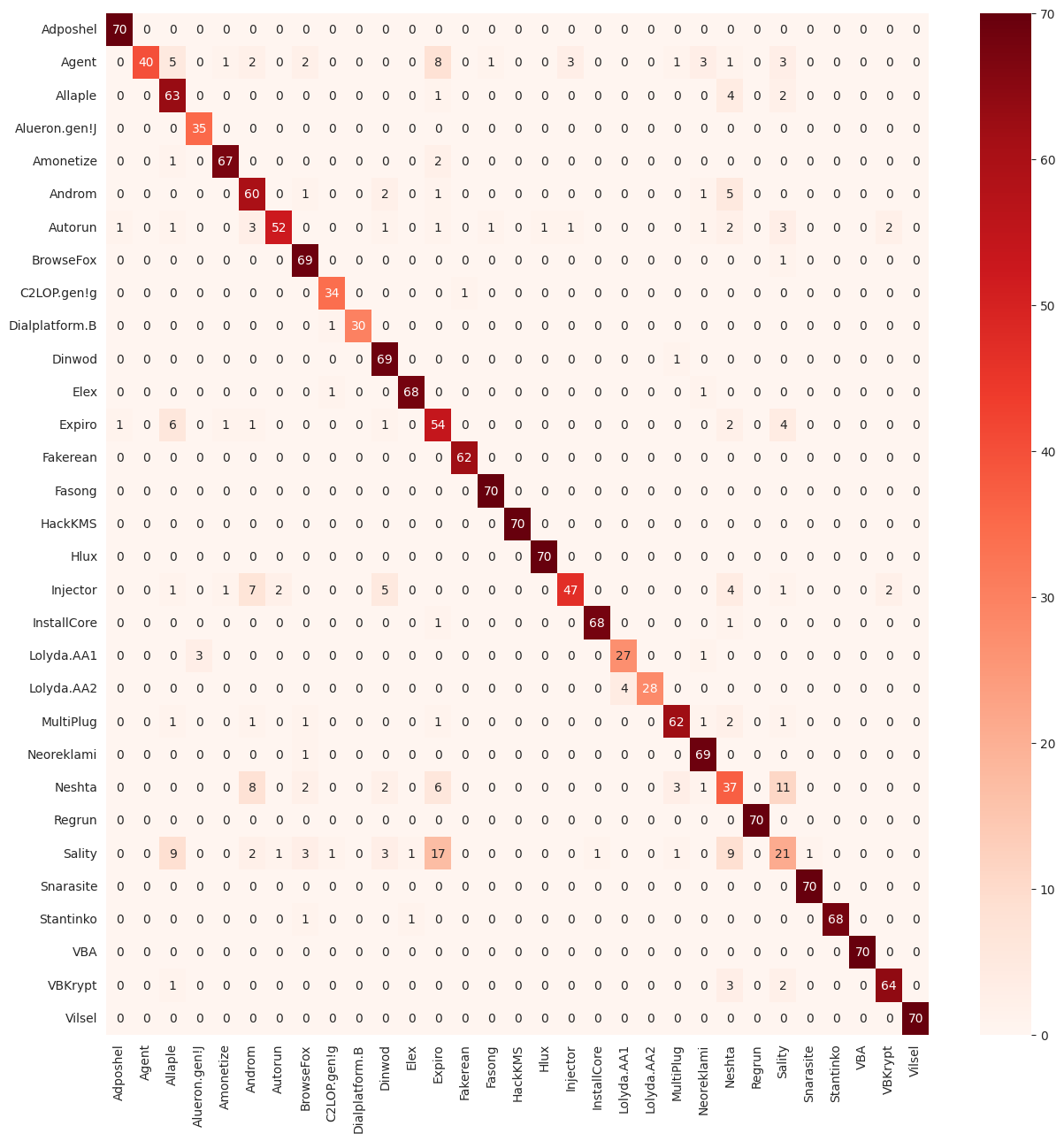
Hình 10. Confusion matrix của XceptionNet với bộ dữ liệu Blended

Hình 11 thể hiện confusion matrix của mô hình EfficientNetB4 với bộ dữ liệu Blended.



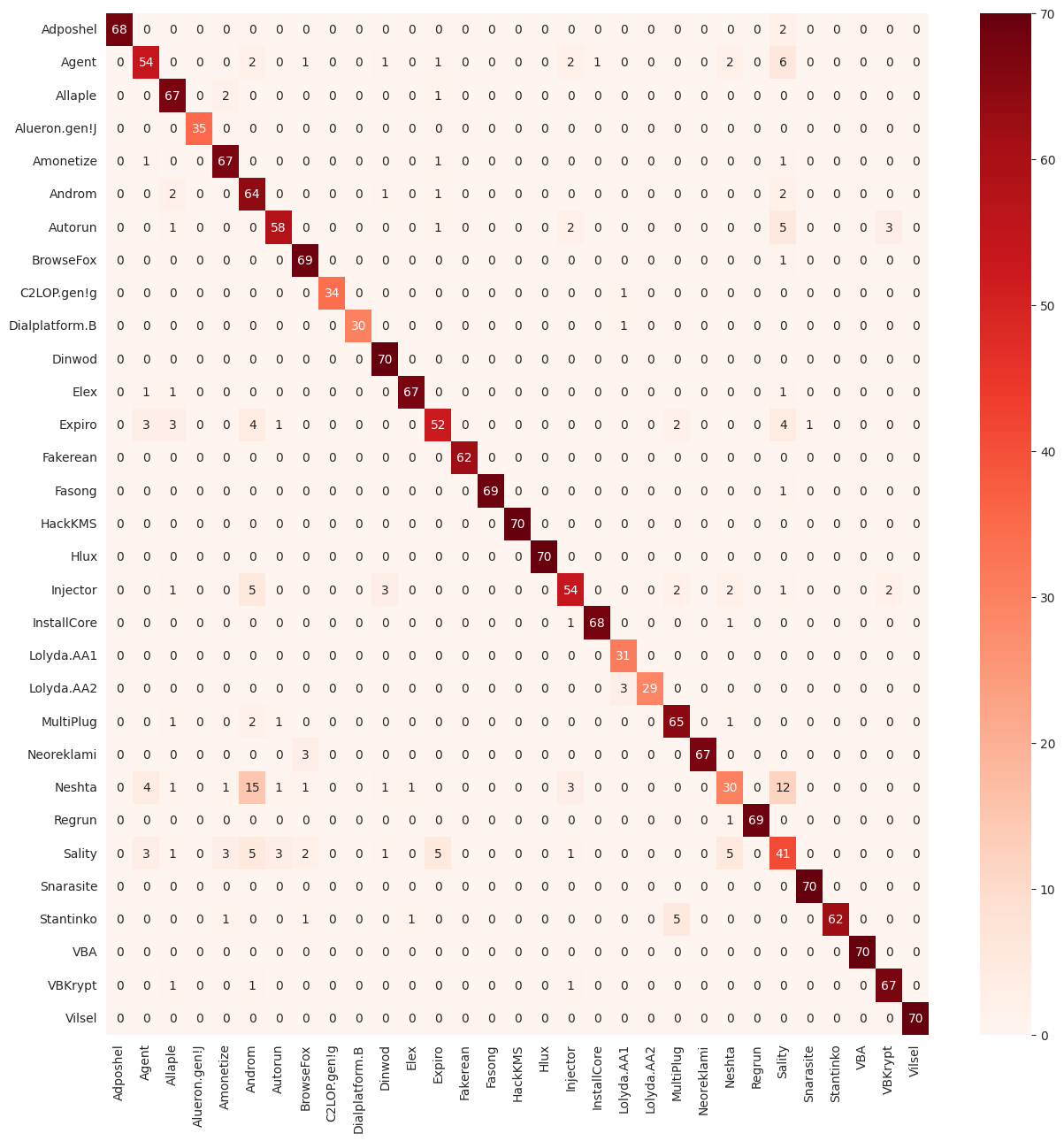
Hình 11. Confusion matrix của EfficientNetB4 với bộ dữ liệu Blended

Hình 12 thể hiện confusion matrix của mô hình ResNet152V2 với bộ dữ liệu Blended.



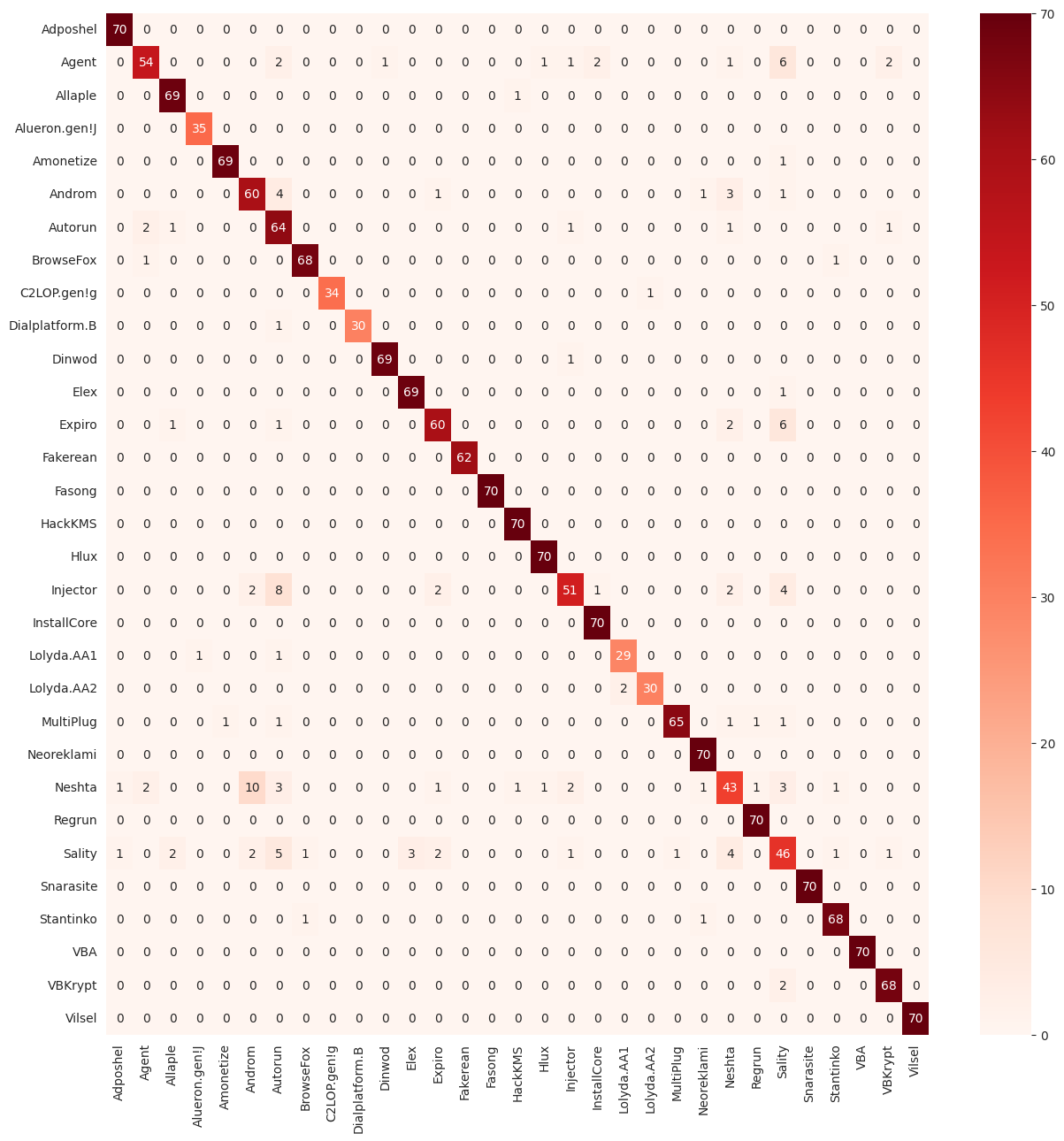
Hình 12. Confusion matrix của ResNet152V2 với bộ dữ liệu Blended

Hình 13 thể hiện confusion matrix của mô hình InceptionV3 với bộ dữ liệu Blended.



Hình 13. Confusion matrix của InceptionV3 với bộ dữ liệu Blended

Hình 14 thể hiện confusion matrix của mô hình DenseNet201 với bộ dữ liệu Blended.

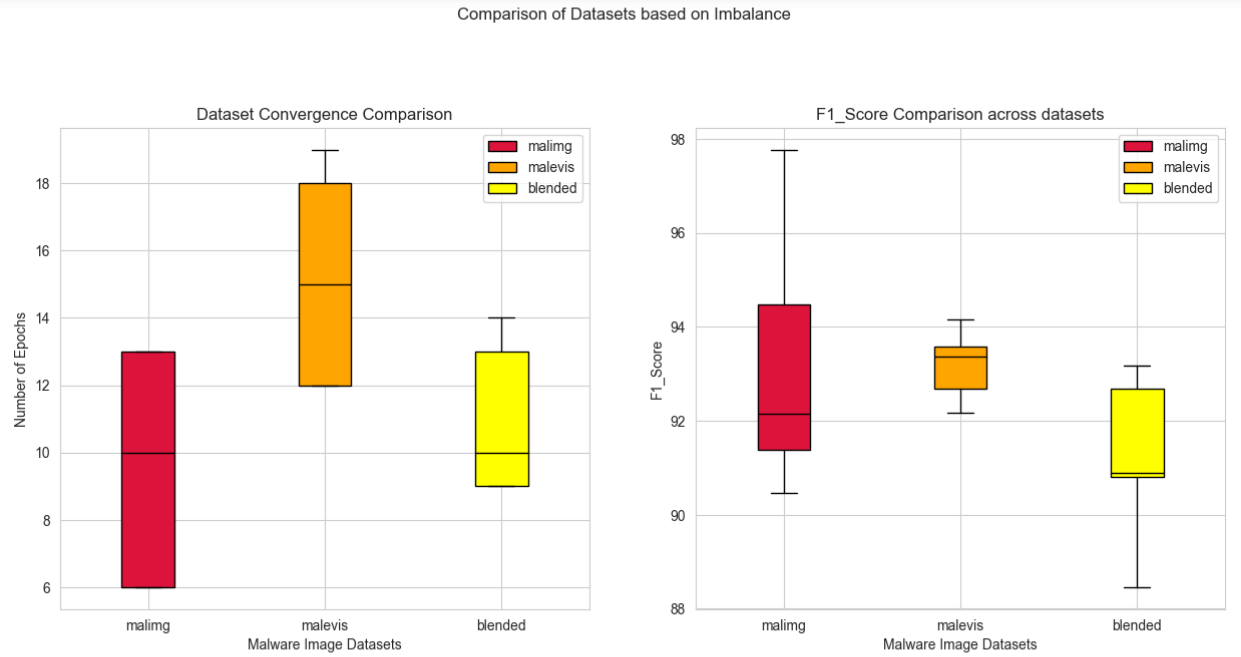


Hình 14. Confusion matrix của DenseNet201 với bộ dữ liệu Blended

Từ các confusion matrix, EfficientNetB4 và DenseNet201 có số lượng các mẫu hình ảnh mã độc PE được phân loại đúng nhiều nhất, mặc dù có nhiều hình ảnh bị phân loại sai nhưng tỉ lệ phân loại sai rất thấp.

## So sánh giữa các bộ dữ liệu

Trước đó, chúng tôi đã so sánh hiệu suất giữa năm mô hình CNN trên từng bộ dữ liệu. Ở phần này, chúng tôi thực hiện so sánh giữa các bộ dữ liệu. Hình 15 biểu diễn biểu đồ hộp về sự hội tụ của mô hình về cơ bản dựa trên sự phân phối số lượng epoch cần thiết của mỗi mô hình và đồng thời khảo sát sự phân phối của F1-score.



Hình 15. So sánh giữa các bộ dữ liệu

Từ biểu đồ hộp có thể thấy rằng bộ dữ liệu Malimg và bộ dữ liệu Blended mất ít epoch nhất để hội tụ trong khi bộ dữ liệu Malevis mất nhiều epoch nhất để hội tụ. Đường trung vị của epoch và F1-score được thể hiện bởi đường gạch ngang trong hộp, nằm hoàn toàn ở trung tâm đối với bộ dữ liệu cân bằng hoàn toàn và lệch nhiều đối với bộ dữ liệu không cân bằng. Thêm vào đó, có sự khác biệt lớn về hiệu suất của các mô hình đối với bộ dữ liệu Malimg so với các bộ dữ liệu khác. Ngoài các chỉ số đánh giá, quy mô và độ phức tạp của mô hình cũng cần được xem xét. Nhìn chung, sự mất cân bằng trong phân bố lớp có ảnh hưởng trực tiếp đến sự hội tụ và hiệu suất của mô hình.

# KẾT LUẬN

Trong nghiên cứu này, chúng tôi đã đề xuất thử nghiệm các mô hình CNN như là XceptionNet, EfficientNetB4, ResNet152V2, InceptionV3 và DenseNet201 cho việc phân loại mã độc PE. Với ba bộ dữ liệu Malimg, Malevis và Blended, chúng tôi đã so sánh hiệu suất của năm mô hình CNN này và nhận thấy rằng mô hình EfficientNetB4 có hiệu suất tốt nhất trên cả ba bộ dữ liệu với chỉ số precision trung bình là 95.25%. Theo sau mô hình EfficientNetB4 là mô hình DenseNet201 với precision trung bình là 94.17%. Các mô hình còn lại cũng có hiệu suất tốt với precision trung bình từ khoảng 91% đến 92%. Kết quả nghiên cứu của chúng tôi cho thấy CNN là một thuật toán hiệu quả để phân loại mã độc PE và trong các mô hình đề xuất thì mô hình EfficientNetB4 là lựa chọn tốt nhất cho nhiệm vụ này. Đồng thời, chúng tôi đã so sánh giữa các bộ dữ liệu và đúc kết rằng mức độ mất cân bằng lớp càng cao trong bộ dữ liệu thì sự khác biệt trong hiệu suất của các mô hình càng lớn và số lượng epoch cần thiết để hội tụ càng ít. Ngoài ra, các mô hình EfficientNetB4, InceptionV3 và DenseNet201 có thể xử lý tốt cả dữ liệu cân bằng và mất cân bằng. Các mô hình còn lại là XceptionNet và ResNet152V2 thì nhạy cảm với sự mất cân bằng lớp. Phân tích so sánh này có thể giúp lựa chọn các mô hình để thử nghiệm khi huấn luyện bất kỳ bộ phân loại mã độc nào.

Về hướng phát triển trong tương lai, chúng tôi sẽ cải thiện hiệu suất của các mô hình CNN bằng phương pháp tinh chỉnh (fine-tuning).

# TÀI LIỆU THAM KHẢO

[1] N. Hưng, “Malware là gì? Có những hình thức Malware nào hiện nay?” 2021. [Online]. Available: https://vietnix.vn/malware-la-gi/. [Accessed: Nov. 17, 2023].

[2] phamcongit, “Tìm hiểu về PE file – P1,” 2017. [Online]. Available: https://phamcongit.wordpress.com/2017/07/06/tim-hieu-ve-pe-file-p1/ [Accessed: Nov. 09, 2023].

[3] AV-TEST, “Malware.” [Online]. Available: https://www.av-test.org/en/statistics/malware/. [Accessed: Nov. 18, 2023].

[4] T. N. Anh and V. K. Lĩnh, “Phát hiện mã độc dựa vào học máy và thông tin PE Header (Phần I),” 2021. [Online]. Available: https://m.antoanthongtin.gov.vn/giai-phap-khac/phat-hien-ma-doc-dua-vao-hoc-may-va-thong-tin-pe-header-phan-i-107476. [Accessed: Nov. 18, 2023].

[5] N. H. M. Khanh, N. V. Chính, and N. P. B. Ngọc, “Sự phát triển một số mạng CNN.” [Online]. Available: https://mmlab.uit.edu.vn/tutorials/ml/deep-learning/mo\_hinh\_cnn\_cai\_tien. [Accessed: Nov. 09, 2023].

[6] T. K. Tran and H. Sato, “NLP-based approaches for malware classification from API sequences,” in 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES), Hanoi, Vietnam, 2017. DOI: 10.1109/IESYS.2017.8233569.

[7] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, “Malware classification with recurrent networks,” in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 2015. DOI: 10.1109/ICASSP.2015.7178304.

[8] M. E. Narayanan, “Malware Classification Using Xgboost With Vote Based Backward Feature Elimination Technique,” in TURCOMAT, 2021. DOI: 10.17762/turcomat.v12i10.5412.

[9] V. L. Duc, N. K. Trong, N. V. Tam, N. N. Tu, M. Fabio, and P. H. Phu, “HIT4Mal: Hybrid image transformation for malware classification,” 2019. DOI: 10.1002/ett.3789.

[10] Y.-H. Chen, J.-L. Chen, and R.-F. Deng, “Similarity-Based Malware Classification Using Graph Neural Networks,” 2022. DOI: 10.3390/app122110837.

[11] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, “Malware Classification with Deep Convolutional Neural Networks,” in 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 2018. DOI: 10.1109/NTMS.2018.8328749.

[12] S. Pai, F. D. Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, “Clustering for malware classification,” Journal of Computer Virology and Hacking Techniques, 2017. DOI: 10.1007/s11416-016-0265-3.

[13] W. Huang and J. W. Stokes, “MtNet: A Multi-Task Neural Network for Dynamic Malware Classification,” in Detection of Intrusions and Malware, and Vulnerability Assessment, 2016. DOI: 10.1007/978-3-319-40667-1\_20.

[14] D. Gibert, C. Mateu, and J. Planes, “HYDRA: A multimodal deep learning framework for malware classification,” 2020. DOI: 10.1016/j.cose.2020.101873.

[15] B. Jung, T. Kim, and E. G. Im, “Malware classification using byte sequence information,” 2018. DOI: 10.1145/3264746.3264775.

[16] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, “Malware Images: Visualization and Automatic Classification,” 2016.

[17] A. S. Bozkir, A. O. Cankaya, and M. Aydos, “Utilization and Comparision of Convolutional Neural Networks in Malware Recognition,” 2019.

[18] G. Pendharkar, “Blended Malware Image Dataset,” 2023.