**1. Introduction**

Các cuộc tấn công ransomware (RW) đã trở thành một trong những mối đe dọa bảo mật lớn nhất mà các cá nhân và doanh nghiệp trên toàn thế giới phải đối mặt. Phần mềm độc hại điển hình nhắm mục tiêu người dùng bằng cách xóa hoặc làm hỏng tệp, thay đổi cấu hình hệ thống, tiết lộ thông tin người dùng cho bên thứ ba, v.v. Mặt khác, ransomware có quyền truy cập vào dữ liệu người dùng và tài nguyên máy tính mà không bị phát hiện, thông báo cho nạn nhân và yêu cầu tiền chuộc để giải phóng quyền truy cập vào các tài nguyên bị bắt (ví dụ: tệp được mã hóa, v.v.). Crypto-ransomware tìm và mã hóa các tệp trên thiết bị bằng mật mã vững chắc để từ chối quyền truy cập của người dùng. Locker ransomware tự khóa thiết bị, chủ yếu bằng cách khóa giao diện người dùng hoặc sử dụng lớp phủ bật lên, để người dùng không thể vào thiết bị ngay từ đầu [1]. Ransomware không chỉ là một hiện tượng hệ điều hành Windows. Nó cũng tấn công các nền tảng khác, chẳng hạn như thiết bị Android. Vào cuối năm 2018, Android chiếm hơn 86,8% tổng thị phần điện thoại di động.1 Android đã nổi lên như là hệ điều hành được sử dụng rộng rãi nhất cho các thiết bị di động [2]. Vào tháng 9 năm 2018, McAfee Lab tuyên bố rằng tổng số ransomware đã lên tới 17 triệu. Phần mềm tống tiền Android sẽ là một trong những mối đe dọa bảo mật quan trọng nhất trong tương lai.2

Phần mềm tống tiền có thể lây lan như một ứng dụng hợp pháp hoặc được tải xuống (vô tình) bởi những người dùng có ý định tải xuống các bản cập nhật phần mềm, ứng dụng, v.v. từ các cửa hàng ứng dụng của bên thứ ba [3] hoặc bằng cách nhấp vào liên kết spam được gửi trong tin nhắn SMS. Tuy nhiên, ransomware Android hiện đại thường lây lan qua các ứng dụng bị xâm nhập có sẵn miễn phí cho người dùng thông qua các cửa hàng ứng dụng của bên thứ ba. Những kẻ tấn công ransomware chọn một ứng dụng phổ biến để bắt chước một ứng dụng thực tế lây nhiễm vào một cơ sở người dùng lớn. Tùy thuộc vào mức độ phức tạp của cuộc tấn công, kẻ tấn công giữ lại chức năng ban đầu của ứng dụng và có thể thêm mã độc vào đó, hoặc ứng dụng chỉ có thể hiển thị biểu tượng và tên của ứng dụng gốc. Điều này được thực hiện để cài đặt ransomware trên thiết bị đích không được chú ý và khiến người dùng [4]. Sau khi cài đặt, ransomware thu thập thông tin về thiết bị của nạn nhân, tìm kiếm các tài nguyên mục tiêu như tệp, tài nguyên, v.v. và liên lạc với máy chủ Command & Control (C &C) để lấy khóa mã hóa nếu nó chưa được bao gồm trong tải trọng. Sau đó, ransomware chiếm quyền điều khiển (khóa / mã hóa) tài nguyên mục tiêu và hiển thị thông báo cho nạn nhân yêu cầu anh ta trả tiền chuộc, cùng với hướng dẫn thanh toán. Hiện tại, ransomware được phát triển chủ yếu cho các thiết bị Android đang gia tăng. Do sự gia tăng đáng báo động của các ứng dụng ransomware Android, phân tích và phát hiện ransomware Android đã trở thành một lĩnh vực nghiên cứu quan trọng. Một số kỹ thuật để phát hiện và phân loại ransomware Android đã được đề xuất. Chúng ta có thể chia các kỹ thuật phát hiện ransomware thành hai loại: Phân tích tĩnh và Phân tích động. Phân tích tĩnh sử dụng cú pháp hoặc thuộc tính cấu trúc của ứng dụng để xác định tính độc hại của nó. Phân tích tĩnh dựa trên trích xuất tính năng (không thực thi) từ các tệp tài nguyên, tệp kê khai Android, mã bytecode Java, v.v. Tệp kê khai Android chứa tất cả các quyền cần thiết, là điểm thiết kế trung tâm của mô hình bảo mật Android [5]. Theo mặc định, không có ứng dụng nào có quyền truy cập dữ liệu nhạy cảm (như danh bạ hoặc SMS) và một số chức năng hệ thống nhất định (như camera, Internet). Các nhà phát triển ransomware sử dụng các quyền chủ yếu để leo thang đặc quyền và truy cập dữ liệu nhạy cảm được lưu trữ trên thiết bị. Phân tích động nhằm mục đích phát hiện hành vi độc hại trong quá trình thực thi chương trình. Phân tích động có thể xem xét các tính năng như tải mã động, chuỗi các cuộc gọi hệ thống được thu thập trong quá trình thực thi ứng dụng, hoạt động mạng, sử dụng CPU và sử dụng bộ nhớ [1]. Sự tương đồng trong hành vi ứng dụng ransomware có thể giúp xác định ransomware mới (zero-day). Hầu hết các kỹ thuật hiện đại [2] không tính đến các đặc điểm cấu trúc cụ thể cho sự xuất hiện của ransomware, chẳng hạn như văn bản trong mã nguồn. Phần mềm tống tiền có thể chứa các mối đe dọa cụ thể trong mã của nó, ví dụ: khóa, mã hóa, khiêu dâm, v.v. Hầu hết các phần mềm tống tiền Android đều yêu cầu các quyền riêng lẻ (chẳng hạn như BIND\_DEVICE\_ADMIN, KILL\_BACKGROUND\_PROCESS và RECEIVE\_BOOT\_COMPLETED, v.v.), có thể hữu ích cho việc phát hiện ransomware. Ransomware thường xuyên thiết lập các kết nối mạng để truy xuất lệnh hoặc gửi dữ liệu được thu thập từ các thiết bị [4]. Do đó, địa chỉ mạng (địa chỉ email, địa chỉ IP, URL) có thể có trong mã của các mẫu ransomware khác nhau, có thể giúp phát hiện ransomware. Các tính năng dựa trên mạng này chưa bao giờ được phân tích tĩnh để phát hiện ransomware Android trước đây.

**1.1. Motivation**

Phân tích hành vi dựa trên các tính năng phần cứng như sử dụng CPU, sử dụng bộ nhớ và nhật ký cuộc gọi hệ thống có thể hữu ích cho việc phân loại ransomware Android, vì chúng có khả năng chống thay đổi cao hơn so với các tính năng tĩnh mà ransomware có thể trốn tránh thông qua xáo trộn mã và mã hóa [1,6,7]. Theo hiểu biết của chúng tôi, việc phát hiện và phân loại (sử dụng máy học) của ransomware Android vẫn chưa được thực hiện bằng cách sử dụng kết hợp các tính năng trên. Do những tiến bộ trong kỹ thuật học máy, một số lượng đáng kể nghiên cứu đã được thực hiện về phát hiện phần mềm độc hại Android bằng cách sử dụng các kỹ thuật học máy [8]. Mặc dù các kỹ thuật học máy đã chứng minh hiệu quả của chúng trong việc phát hiện phần mềm độc hại, nhưng các trình phân loại học máy không có khả năng phục hồi tốt trước các cuộc tấn công đối thủ. Khía cạnh này được nhấn mạnh trong văn bản sau: ''trong giai đoạn học tập, tập dữ liệu được sử dụng để đào tạo vẫn đại diện cho miền vấn đề, giả sử không có sửa đổi độc hại có chủ ý đối với dữ liệu'' [9]. Do đó, người dùng độc hại thường sử dụng các cuộc tấn công đối nghịch để đánh lừa các mô hình học máy. Chúng ta có thể chia các cuộc tấn công đối nghịch thành hai loại: (1) trốn tránh và (2) tấn công đầu độc [9,10]. Trong các cuộc tấn công trốn tránh, kẻ tấn công cố tình chế tạo các đầu vào độc hại để mô hình phân loại không chính xác phân loại một ứng dụng là lành tính hoặc sạch sẽ. Mặt khác, trong các cuộc tấn công đầu độc, kẻ tấn công đầu độc dữ liệu đào tạo để làm tổn hại toàn bộ quá trình học tập. Trọng tâm của công việc này là chống lại các cuộc tấn công vượt kiểm duyệt của kẻ thù. Chúng tôi xác định một tập hợp con các tính năng từ một số tính năng của các loại hành vi khác nhau. Hơn nữa, chúng tôi xây dựng tập hợp nhiều tính năng riêng biệt thay vì sử dụng một vectơ tính năng duy nhất. Động lực chính để sử dụng một vectơ tính năng duy nhất dựa trên một số tập con riêng biệt là khái quát hóa. Điều này giúp các trường hợp đóng góp nhiều hơn vào các phân loại được đào tạo. Bằng cách này, kẻ tấn công khó có thể vượt qua mô hình phát hiện. Chúng tôi sử dụng phần mềm tống tiền Android chuẩn để trích xuất nhiều tập hợp con phân biệt đối xử dựa trên phân tích hành vi của chúng. Chúng tôi đã sử dụng một tập hợp nhiều bộ phân loại và kết hợp chúng với phương pháp bỏ phiếu. Cuối cùng, chúng tôi đã so sánh phương pháp được đề xuất với mô hình dựa trên ML truyền thống trong môi trường đối nghịch.

**1.2. Contributions**

Nghiên cứu này tập trung vào việc giảm thiểu các cuộc tấn công trốn tránh trong phát hiện ransomware Android, chẳng hạn như xáo trộn mã và sử dụng nó để tránh phát hiện phần mềm độc hại / ransomware. Hầu hết các kỹ thuật ransomware hiện tại không sửa đổi vectơ tính năng đầu vào để phân tích (khi một khía cạnh được sử dụng để xáo trộn, nó sẽ thay đổi toàn bộ vectơ tính năng, khiến trình phân loại được đào tạo phân loại sai ransomware. Do đó, nghiên cứu này đề xuất một cơ chế phân tích dựa trên tổng hợp để phát hiện ransomware Android và giảm thiểu các cuộc tấn công trốn tránh. Các đặc điểm Android được sử dụng để phân tích ở đây không dễ sửa đổi và sử dụng cho các nỗ lực trốn tránh. Dựa trên động lực này, chúng tôi đề xuất một kỹ thuật kết hợp hiệu quả của cả tính năng tĩnh (ví dụ: quyền, văn bản, tính năng dựa trên mạng, v.v.) và các tính năng động (như nhật ký cuộc gọi hệ thống, CPU và sử dụng bộ nhớ) để phát hiện phần mềm tống tiền Android bằng cách sử dụng mô hình máy học tổng hợp. Tóm lại, những đóng góp chính của nghiên cứu này là:

1. Trích xuất và phân tích các tính năng dựa trên mạng tĩnh như địa chỉ IP, địa chỉ email và URL.

2. Phát triển hai mô hình tập hợp máy học khác nhau bao gồm nhiều thuật toán học máy cho các bộ tính năng tĩnh và động để phát hiện mã độc tống tiền Android và giảm thiểu các cuộc tấn công trốn tránh đối thủ.

3. Đánh giá hiệu quả của mô hình được đề xuất để giảm thiểu các cuộc tấn công trốn tránh đối thủ bằng cách sử dụng một bộ dữ liệu lớn các vectơ tính năng được chế tạo từ các mẫu ransomware Android.

Chúng tôi đã cấu trúc phần còn lại của bài báo như sau. Trong Phần 2 tiếp theo, chúng tôi trình bày một đánh giá tài liệu về các kỹ thuật liên quan. Phần 3 mô tả phương pháp được đề xuất, tiếp theo là Phần 4, nơi thiết lập thử nghiệm được trình bày. Phần 5 cung cấp một đánh giá chi tiết và thảo luận về các kết quả thu được và Phần 6 kết luận bài báo.

**2. Related work**

Có một số cách tiếp cận để phát hiện ransomware và một số trong số chúng liên quan đến nền tảng Android. Song et al. [2] đã đề xuất một kỹ thuật giám sát cẩn thận và chỉ định các quy trình và thư mục tệp nhất định bằng cách sử dụng số liệu thống kê về việc sử dụng bộ xử lý, sử dụng bộ nhớ và tốc độ I / O để có thể phát hiện các quy trình có hành vi bất thường. Kỹ thuật này đã được thực hiện với ba mô-đun (tức là cấu hình, giám sát và xử lý). Mô-đun cấu hình tạo các bảng danh sách giám sát, trong khi mô-đun giám sát giám sát bộ xử lý, bộ nhớ và việc sử dụng I / O của từng quy trình. Mô-đun xử lý xử lý xử lý các quy trình được mô-đun giám sát coi là đáng ngờ và đưa ra ngoại lệ hoặc cô lập chúng. Cách tiếp cận được đề xuất có thể được thực hiện trong mã nguồn Android. Nếu không có được thông tin về ransomware, nó có thể làm giảm thiệt hại do ransomware không xác định gây ra. Tuy nhiên, kỹ thuật này không thực hiện phân tích tĩnh, có thể hiệu quả hơn. Yang et al. minh họa thiết kế của một kỹ thuật phân tích lai tự động [3]. Hệ thống được đề xuất sử dụng phân tích tĩnh dựa trên các tính năng phù hợp như quyền, trình tự gọi API, tài nguyên và cấu trúc APK. Phân tích động mô tả bản chất của cuộc tấn công dưới dạng rò rỉ dữ liệu như cookie trình duyệt web và các loại khác mà không có quyền truy cập vào các nguồn dữ liệu được bảo vệ chính xác trong thiết bị di động. Alberto et al. đã đề xuất một cách tiếp cận lai để phát hiện ransomware Android, trước tiên kiểm tra (sử dụng phân tích tĩnh dựa trên tần số opcode) ứng dụng sẽ được sử dụng trên thiết bị trước khi cài đặt [8]. Sau đó, phân tích động xác định xem hệ thống có bị tấn công hay không bằng cách theo dõi việc sử dụng CPU, sử dụng bộ nhớ, sử dụng mạng và thống kê cuộc gọi hệ thống. Bộ dữ liệu được sử dụng cho các thí nghiệm là nhỏ và phân tích như vậy không thể kiểm tra các họ ransomware Android khác nhau.

Học tập dựa trên đồng bộ giúp cải thiện các biện pháp đối phó trong môi trường đối nghịch [9]. Một thế hệ tấn công dựa trên phát sinh mà không thực hiện chức năng độc hại đã được điều tra. Các khởi tạo mới dựa trên các ví dụ đối nghịch được sử dụng như một ví dụ để đào tạo. Điều này cải thiện phương pháp hòa tấu và có thể được sử dụng như một mô hình phân loại mạnh mẽ hơn. Các phương pháp dựa trên đồng bộ được sử dụng để phân loại duy nhất và phát hiện phản hồi giả mạo

Gharib et al. [12] đã đề xuất một kỹ thuật droid DNA, một khung phát hiện thời gian thực lai có thể nhanh chóng đánh giá một mẫu bằng cách sử dụng phân tích tĩnh. Nếu ứng dụng chỉ được coi là đáng ngờ, nó sẽ được theo dõi liên tục và hành vi thời gian chạy được lập hồ sơ. Khi hồ sơ giống như một tập hợp các hồ sơ độc hại, DNA-Droid sẽ chấm dứt ứng dụng. Kiến trúc tổng thể của khung đề xuất của họ bao gồm ba thành phần chính: phân tích tĩnh, phân tích động và mô-đun phát hiện. Mô-đun tĩnh bao gồm ba thành phần con (Mô-đun phân loại văn bản (TCM), Mô-đun phân loại hình ảnh (ICM) và Mô-đun quyền và lệnh gọi API (APM)) để đánh giá các khía cạnh khác nhau của tệp APK. Mô-đun động cấu hình các họ phần mềm độc hại dựa trên chuỗi lệnh gọi API và tạo DNA cho từng họ. Trong giai đoạn phát hiện, hành vi thời gian chạy của một mẫu đáng ngờ liên tục được so sánh với các họ của DNA. Alzahrani et al. [13] đã giới thiệu Randroid. Cách tiếp cận tự động này đo lường sự tương đồng về cấu trúc giữa thông tin được thu thập của ứng dụng được kiểm tra và thông tin liên quan đến mối đe dọa được thu thập bởi các biến thể ransomware đã biết để phân loại ứng dụng là ransomware hoặc goodware. Cách tiếp cận Randroid trích xuất thông tin của ứng dụng như hình ảnh và văn bản từ các tệp bố cục XML, tài nguyên và tệp class.dex trong giai đoạn phân tích tĩnh. Phân tích động nắm bắt hoạt động tống tiền và kiểm tra sự hiện diện của các thư đe dọa hoặc màn hình khóa. Đo lường tương tự hình ảnh (ISM) và Đo lường tương tự chuỗi (SSM) được sử dụng để xác định sự giống nhau giữa thông tin được trích xuất và thông tin được thu thập trước đó về ransomware đã biết. Dựa trên điểm số tương tự, ứng dụng được kiểm tra được phân loại là đáng ngờ, phần mềm tốt hoặc ransomware. Một giải pháp được đề xuất khác để phát hiện ransomware tiền điện tử đã được đề xuất bởi Chen et al. [14], cụ thể là RansomProber. Ran somProber là một kỹ thuật phát hiện thời gian thực phân tích các tiện ích Giao diện người dùng (UI) của các hoạt động liên quan, điều phối chuyển động ngón tay của người dùng và phát hiện xem ransomware có bắt đầu quá trình mã hóa tệp hay không. Ransom prober bao gồm ba bước: phân tích mã hóa, phân tích tiền cảnh và phân tích bố cục. Mô-đun phân tích mã hóa được sử dụng để phát hiện xem một số tệp có được mã hóa hay không. Mô-đun phân tích nền trước quyết định xem hoạt động mã hóa có thuộc về ứng dụng của người dùng hay không và mô-đun phân tích bố cục phân tích các tiện ích giao diện người dùng của các hoạt động tương ứng và tọa độ của người dùng. Ransom prober có thể phát hiện ransomware đóng gói lại nhắm mục tiêu vào một ứng dụng mà không cần mã hóa. Kỹ thuật này chỉ được thiết kế để phát hiện mã độc tống tiền. Cách tiếp cận được đề xuất không phát hiện các ứng dụng được đóng gói lại sử dụng phương pháp mã hóa hoặc nén.

Mercaldo et al. đã mô tả một kỹ thuật kiểm tra mô hình để xác định tải trọng độc hại trong ransomware Android [15]. Kỹ thuật này được chia thành ba quy trình con (xây dựng mô hình chính thức, xây dựng các thuộc tính logic thời gian và phát hiện họ ransomware). Trong việc xây dựng mô hình chính thức, bytecode của ứng dụng được phân tích cú pháp và các mô hình chính thức thích hợp của hệ thống được tạo. Việc xây dựng Thuộc tính logic thời gian xác định hành vi đặc trưng của ransomware dưới dạng một tập hợp các thuộc tính. Trong phát hiện họ ransomware, môi trường xác minh chính thức bao gồm trình kiểm tra mô hình được sử dụng để phát hiện các gia đình ransomware. Các tính năng đặc biệt của phương pháp này là các phương pháp chính thức và phát hiện ransomware từ mã byte Java.

A white sheet with black text

Description automatically generated with medium confidence

Các phương pháp dựa trên tập hợp đã được áp dụng theo một số cách tiếp cận [16]. Phương pháp này kết hợp học tập tổng hợp và học chuyển giao trên dữ liệu chuỗi thời gian để thực hiện các cập nhật gia tăng. Kết quả là, phương pháp lai đưa ra dự đoán một cách độc lập và cải thiện độ chính xác. Trong một nghiên cứu khác, NATICUSdroid [17] đã được đề xuất, trong đó chọn các quyền cụ thể làm tính năng và sau đó phân loại chúng là lành tính hoặc phần mềm độc hại. Việc lựa chọn dựa trên xu hướng của các quyền. Mô hình đề xuất được đánh giá bằng tám thuật toán học máy khác nhau. Mô hình phân loại rừng ngẫu nhiên đạt độ chính xác 97%, tỷ lệ dương tính giả là 3,32% và độ đo F là 0,96.

Một phương pháp khác PerbDroid [18] đã được đề xuất bằng cách sử dụng sáu cách tiếp cận khác nhau để xếp hạng tính năng (ví dụ: tỷ lệ khuếch đại, đánh giá tính năng OneR, kiểm tra chi bình phương, đánh giá tính năng thu được thông tin, phân tích thành phần chính (PCA) và phân tích hồi quy logistic) để tạo ra thuật toán phân loại dựa trên API, quyền và ý định. Mô hình đã được thử nghiệm với ứng dụng thực tế [18]. Mô hình đã sử dụng thuật toán phân loại và quy trình lựa chọn siêu điều chỉnh để chọn và phân loại khả năng phát hiện phần mềm độc hại. Tuy nhiên, việc phát hiện phần mềm độc hại và các biện pháp đối phó dựa trên các mô hình đối thủ vẫn cần được khám phá.

Chúng tôi tóm tắt điểm mạnh và điểm yếu của các phương pháp hiện tại trong Bảng 1. Đánh giá tài liệu cho thấy hầu hết các kỹ thuật được đề xuất cho đến nay, cho dù chúng chỉ thực hiện phát hiện tĩnh hay động [2]. Mặc dù phân tích tĩnh nhanh, an toàn và chính xác trong việc xác định các mẫu ransomware đã biết [1], nhưng tính năng phát hiện chỉ tĩnh có thể dễ bị tấn công bởi ransomware làm xáo trộn mã để thay đổi cấu trúc [15] và không thể xử lý các mẫu mã hóa hoặc nén chúng. tải trọng. Phân tích động có khả năng chống trốn tránh [8]. Nó có thể phát hiện phần mềm ransomware không xác định dựa trên các dấu hiệu hành vi chung [2]. Ngoài ra, phân tích động có một số lỗ hổng, chẳng hạn như một số hành động chỉ kích hoạt một số điều kiện nhất định có thể không có trong môi trường thử nghiệm, chẳng hạn như trình mô phỏng [1]. Do đó, để phát hiện ransomware, có thể cần kết hợp khả năng phát hiện dựa trên hành vi có khả năng chống trốn tránh [2] với khả năng phân tích tĩnh hiệu quả [12]. Một số kỹ thuật khác sử dụng phương pháp kết hợp là dành riêng cho từng loại, chỉ giải quyết một loại, chẳng hạn như phần mềm ransomware tiền điện tử [14] hoặc chỉ giải quyết một họ ransomware cụ thể [3]. Tính năng phát hiện dành riêng cho từng dòng không thể khái quát hóa giải pháp và áp dụng nó cho mọi loại ransomware. Các phương pháp được đề xuất trước đây để phát hiện phần mềm tống tiền trên Android [8] bằng cách sử dụng kỹ thuật học máy rất dễ bị tấn công lẩn tránh đối nghịch. Những kẻ tấn công có thể xâm phạm toàn bộ mô hình phát hiện bằng cách làm xáo trộn mã hoặc sử dụng các kỹ thuật trốn tránh khác, vì việc thay đổi một khía cạnh sẽ thay đổi toàn bộ vectơ đặc trưng; do đó, phần mềm tống tiền vẫn không bị bộ phân loại học máy phát hiện.

Nhìn chung, công việc này đã truyền cảm hứng cho chúng tôi đề xuất một kỹ thuật phân loại ransomware Android dựa trên phân tích kết hợp, sử dụng cả tính năng tĩnh và động hiệu quả cũng như học máy dựa trên học tập tổng hợp để giảm thiểu cuộc tấn công lẩn tránh đối thủ.

**3.Hybrid distinct ensemble analyze**

Chúng tôi đã thảo luận về khung đề xuất để phân loại ransomware trong Hình 1. Phương pháp được đề xuất bao gồm hai phần, tức là đào tạo ngoại tuyến và dự đoán trực tuyến, như trong Hình 1 (a) và (b). Trong phần đào tạo ngoại tuyến, việc trích xuất, lựa chọn, đào tạo và kiểm tra tính năng được thực hiện. Trong phương pháp dự đoán trực tuyến, bộ phân loại đã được đào tạo được sử dụng để dự đoán các ứng dụng ransomware và không phải ransomware. Trích xuất tính năng dựa trên phân tích tĩnh và động được thảo luận trong 3.1 và 3.2. Trong đào tạo ngoại tuyến, phân tích tĩnh được thực hiện để trích xuất quyền, mạng và văn bản như trong Hình (a). Sau đó, các bộ phân loại riêng biệt sẽ được huấn luyện cho từng loại đối tượng địa lý. Bằng cách này, bộ phân loại meta tĩnh được đào tạo dựa trên quyền, tính năng mạng và văn bản. Trong phân tích động, các ứng dụng được chạy trong môi trường được kiểm soát, sau đó các tính năng được trích xuất và phân tích. Chúng tôi đã ghi lại các lệnh gọi hệ thống, mức sử dụng CPU và mức tiêu thụ bộ nhớ của ứng dụng và sử dụng chúng làm tính năng. Các mô hình phân loại riêng biệt sau đó được đào tạo tùy thuộc vào bản chất của các tính năng. Bằng cách này, một bộ phân loại meta dựa trên tính năng động sẽ được huấn luyện. Phân tích tập hợp lai được đề xuất bao gồm hai mô hình học máy tập hợp riêng biệt được huấn luyện trên các vectơ đặc trưng tĩnh và động. Mỗi mô hình tập hợp bao gồm một số lẻ các thuật toán phân loại học máy, ví dụ: ba hoặc năm. Phương pháp tập hợp xếp chồng được sử dụng để huấn luyện và kiểm tra các thuật toán phân loại, trong đó mỗi thuật toán được huấn luyện trên toàn bộ vectơ đặc trưng. Trong phương pháp dự đoán ngoại tuyến, các bộ phân loại meta tĩnh và động sau đó được sử dụng trong phương thức bỏ phiếu để dự đoán kết quả đầu ra, như đã đề cập trong Hình 1(b). Do các mô hình học tập tổng hợp riêng biệt được sử dụng trong phương pháp luận, chúng tôi gọi nó là một máy phân tích tổng hợp riêng biệt lai.

Chúng tôi bắt đầu với tập dữ liệu gồm các tệp APK phần mềm độc hại chứa cả phần mềm ransomware Android và phần mềm không phải ransomware làm đầu vào. Mỗi ứng dụng Android được đóng gói trong một tệp .apk, một tệp nén chứa một số tệp và thư mục khác, chẳng hạn như tệp class-dex, nội dung, tài nguyên, tệp META-INF và AndroidManifest.xml, v.v. Trong giai đoạn đầu tiên, các tính năng được sử dụng để phân tích tĩnh và động được trích xuất từ tệp APK. Sau đó, các đặc trưng được trích xuất và việc biên dịch chúng thành các vectơ đặc trưng sẽ được sử dụng để thực hiện đồng thời phân tích tĩnh và động. Để tríc

xuất các tính năng tĩnh, tệp APK được phân tách thành các tệp Java và XML. Vì điều nàyA diagram of a train system

Description automatically generated

Công cụ Apk là một tiện ích mã nguồn mở miễn phí giải nén tệp APK vào các tài nguyên riêng lẻ của nó [19]. Các tệp Java và XML thu được được quét thêm để trích xuất các tính năng. Android có một chiến lược cấp phép đặc biệt. Người dùng cấp quyền khi ứng dụng được cài đặt [12]. Các quyền này được trích xuất từ các tệp AndroidManifest.xml. Trong khi các tính năng dựa trên văn bản và mạng như địa chỉ email, địa chỉ IP và URL được trích xuất từ các tệp Java. Các tính năng mạng này mô tả ứng dụng giao tiếp với ai sau khi cài đặt. Vì các hoạt động ransomware Android chủ yếu dựa trên mạng [4], các tính năng mạng này giúp phát hiện ransomware Android. Các tính năng này được chuyển đổi thành một vectơ tính năng kết hợp và được đưa vào mô hình học máy tập hợp tĩnh được đề xuất trong giai đoạn thứ hai. Mô hình học máy tập hợp tĩnh được đào tạo bằng cách sử dụng các vectơ tính năng này. Khi tập hợp tĩnh này được đào tạo, nó có thể phân loại ứng dụng và gán nhãn RW / NRW dựa trên các tính năng tĩnh giống hệt nhau. Tương tự, mỗi tệp APK được chạy trong môi trường mô phỏng để ghi lại các tính năng động để phân tích động. Các tính năng động được trích xuất của tệp APK được chuyển đổi thành vectơ tính năng. Các vectơ tính năng này được sử dụng thêm để đào tạo mô hình hòa tấu động. Sau đó, nhóm độn`‘‘‘```g phân loại ứng dụng và gán nhãn RW / NRW. Quyết định cuối cùng (về phân loại ứng dụng) có tính đến phản hồi từ cả mô hình đồng bộ tĩnh và động. Giả sử một trong các mô hình học máy (một phần của khung tập hợp) phân loại ứng dụng là ransomware (bằng cách gán cho nó nhãn RW). Trong trường hợp này, ứng dụng được phân loại là ransomware. Ứng dụng này chỉ được phân loại là không ransomware nếu cả hai mô hình tổng hợp tĩnh và động gán cho nó một nhãn tương tự (tức là NRW không ransomware).

**3.1. Static feature extractor**

Bộ dữ liệu thử nghiệm của chúng tôi bao gồm các tệp .apk (tức là 50% phần mềm tống tiền và 50% không phải phần mềm tống tiền). Bộ gói Android (APK) là định dạng tệp mà Android sử dụng để phân phối và cài đặt ứng dụng. Nó chứa tất cả các yếu tố như lớp (tệp .dex), tài nguyên và tệp kê khai mà ứng dụng cần được cài đặt chính xác trên thiết bị. Tệp kê khai chứa các quyền và các chi tiết cấu hình khác của ứng dụng. Quá trình trích xuất tính năng của chúng tôi bắt đầu bằng việc chụp các tệp APK bằng tập lệnh trích xuất tính năng. Chúng tôi đã viết một tập lệnh Python để trích xuất quyền từ tệp manifest.xml, văn bản và các tính năng dựa trên mạng (ví dụ: địa chỉ IP, địa chỉ email và URL) từ các tệp .dex. Tập lệnh dịch ngược các tệp APK, trích xuất các tính năng này và sau đó lưu chúng vào các tệp văn bản. Chúng tôi sử dụng các tệp .txt này từ cả ứng dụng ransomware và không phải ransomware để tạo vectơ tính năng. Tập lệnh vector tính năng đọc các tệp .txt từ cả ứng dụng ransomware và không ransomware. Nó xuất ra các vectơ tính năng của mỗi ứng dụng sau khi chụp một tập dữ liệu của tất cả các tính năng để xác định các tính năng đặc trưng của từng ứng dụng và lưu tập dữ liệu trong tệp đầu ra. Chúng tôi tạo một chuỗi nhị phân cho mỗi ứng dụng tại vị trí của nó trong tập dữ liệu (đại diện cho vectơ tính năng). Tất cả các quyền riêng lẻ được phát hiện sau đó được sắp xếp theo thứ tự 0 và 1. Một ủy quyền cụ thể được ký hiệu bằng một ủy quyền và sự vắng mặt của ủy quyền được ký hiệu bằng 0 trong danh sách. Bit cuối cùng của vectơ đại diện cho danh mục của ứng dụng (tức là ransomware hoặc không ransomware). Tất cả các quyền dư thừa sẽ bị xóa khỏi tập dữ liệu, vì sự dư thừa có thể có tác động tiêu cực đến việc phân loại. Sau khi xóa các quyền dư thừa, chúng tôi nhận được 166 quyền duy nhất. Cả hai tính năng dựa trên văn bản và mạng đều chứa chuỗi; do đó, chúng tôi tạo các vectơ tính năng của chúng bằng cách sử dụng vectơ TF-IDF [20]. Bộ vector TF-IDF chuyển đổi các tính năng văn bản thành các vectơ tính năng có thể được sử dụng làm đầu vào cho thuật toán phân loại. TF IDF là một đại diện xa hoa và ngày càng hiệu quả cho các cơ chế phân loại dữ liệu văn bản [21]. Tiếp theo, chúng tôi sử dụng tất cả các vectơ tính năng tĩnh đã tạo để đào tạo các máy phân tích quần thể tĩnh dựa trên học máy. Thuật toán 1 mô tả quá trình trích xuất tính năng cho cả phân tích tĩnh và động của tệp APK, chuyển đổi các tính năng được trích xuất thành vectơ tính năng và cơ chế phân loại được sử dụng để phát hiện ransomware Android.

A close-up of a document

Description automatically generated

**3.2. Dynamic feature extraction**

Phân tích động liên quan đến việc chạy một ứng dụng Android trong môi trường ảo để kiểm tra hành vi thời gian chạy của nó. Phân tích động có thể được sử dụng để phát hiện hành vi độc hại của các ứng dụng vẫn không bị phát hiện trong phân tích tĩnh. Trong nghiên cứu trước đây liên quan đến phân tích động, người ta đã chú ý nhiều đến việc điều tra rò rỉ dữ liệu và chuỗi chuỗi lệnh gọi API [3,12]. Một cách tiếp cận đầy hứa hẹn để phát hiện động hiệu quả ransomware Android là xác định một bộ tính năng hữu ích cho phép phân biệt giữa hành vi ransomware và không ransomware. Khung đề xuất của chúng tôi khám phá các loại tính năng động như vậy (ví dụ: sử dụng CPU, thống kê cuộc gọi hệ thống, sử dụng bộ nhớ, v.v.) ít tốn kém hơn và nhiều thông tin hơn để phát hiện mã độc tống tiền Android [8]. Do đó, dấu vết thực thi chứa dữ liệu này phải được thu thập bằng cách chạy ứng dụng trong một môi trường được kiểm soát. Những dấu vết này được ghi lại thủ công bằng cách chạy từng ứng dụng một trong 10 phút trong trình giả lập Android. Tuy nhiên, một số dấu vết ngắn hơn vì trình giả lập có những điểm yếu nhỏ. Tuy nhiên, thời gian thực hiện dài hơn mang lại cho chúng ta kết quả có ý nghĩa hơn. Trình giả lập Android Genymotion phiên bản 3.0.2 đã được chọn để phân tích động các ứng dụng phần mềm độc hại Android. Công cụ Genymotion được sử dụng vì nó là phần mềm mã nguồn mở và hỗ trợ Android Studio. Lý do sử dụng phần mềm giả lập Android thay vì thiết bị gốc là môi trường mô phỏng cung cấp nhiều dung lượng hơn để chạy một số lượng lớn các chương trình phần mềm độc hại trong một thời gian hợp lý. Phân tích động trên thiết bị miễn nhiễm với các kỹ thuật bỏ qua trình giả lập. Tuy nhiên, trong trường hợp ransomware Android, thiết bị vật lý không thể được đặt lại về trạng thái sạch. Ngược lại, trình giả lập có thể được khởi tạo lại sau khi phân tích từng ứng dụng. Tuy nhiên, việc chống mô phỏng có thể có một số tác động đến việc trích xuất tính năng động được thực hiện trên Genymotion. Do đó, trong mô hình đề xuất của chúng tôi, khi sử dụng các tính năng động (sử dụng bộ nhớ và CPU), chúng tôi giả định rằng ngay cả khi các ứng dụng được đóng gói lại, làm xáo trộn hoặc được trang bị các kỹ thuật để tránh lỗi phát hiện, chúng vẫn sẽ hiển thị các dấu vết hành vi tương tự trong quá trình thực thi. Một trình phân tích tổng hợp được đào tạo về các đặc điểm này có thể phân biệt đầy đủ các ứng dụng ransomware Android với các ứng dụng phần mềm độc hại khác. Thiết bị ảo được khởi tạo lại mỗi lần trước khi một ứng dụng độc hại mới được thực thi để tránh sự can thiệp từ các ứng dụng đã thực thi trước đó, chẳng hạn như thay đổi cài đặt, thực thi các quy trình nền, thay đổi liên quan đến cấu hình hệ điều hành, v.v. Android Debug Bridge (ADB) được sử dụng để giám sát việc sử dụng bộ nhớ và CPU của các ứng dụng. ADB là một công cụ dòng lệnh cho phép PC giao tiếp với phiên bản giả lập hoặc thiết bị Android. Strace (một công cụ theo dõi cuộc gọi hệ thống) được sử dụng để thu thập các cuộc gọi hệ thống từ các ứng dụng. Để trích xuất tính năng động, các bước sau được thực hiện cho từng ứng dụng như được đề cập trong Thuật toán 2.

A screenshot of a computer

Description automatically generated

Chúng tôi đã xem xét tất cả các tính năng liên quan đến cuộc gọi hệ thống, bộ nhớ và việc sử dụng CPU có thể được truy cập trong Android. Tổng cộng, có 73 tính năng cho mỗi ứng dụng đang chạy. Năm tính năng liên quan đến CPU: ba tính năng sử dụng CPU và hai tính năng ngoại lệ bộ nhớ ảo (lỗi lớn và nhỏ). 63 tính năng liên quan đến các khía cạnh khác nhau của việc sử dụng bộ nhớ và 5 tính năng đại diện cho số liệu thống kê về các cuộc gọi hệ thống. Các tính năng này được chuyển đổi thành các vectơ tính năng (tức là các giá trị số) để phân loại. Các vectơ tính năng và danh mục ứng dụng này (tức là ransomware/non-ransomware) được sử dụng để đào tạo các máy phân tích đồng bộ động dựa trên máy học.

**3.3. Ensemble learning**

Hai mô hình học máy riêng biệt được sử dụng để phân loại các ứng dụng dựa trên các tính năng tĩnh và động của chúng. Mỗi nhóm được đào tạo để cung cấp các vectơ tính năng dọc theo danh mục (tức là ransomware có giá trị 1 và không ransomware có giá trị bằng 0) cho các mô hình đồng bộ. Mỗi bộ phân loại (ví dụ: Naïve Bayes, Decision Tree, Random forest, v.v.) trong mô hình tổng hợp được đào tạo với tất cả các vectơ tính năng. Khi tất cả các mô hình hòa tấu này được đào tạo, chúng có thể phân loại các ứng dụng và gán nhãn lớp như RW / NRW. Tất cả các trình phân loại thành viên được cung cấp cho một trình phân loại meta, kết hợp các kết quả này bằng cách sử dụng quy tắc kết hợp (tức là bỏ phiếu theo đa số) để gán nhãn cuối cùng. Vì chúng tôi đang giải quyết các vấn đề hai lớp, chúng tôi sử dụng sơ đồ quyết định đa số để xác định nhãn cuối cùng. Dựa trên kết quả của cả hai mô hình đồng bộ, nhãn cuối cùng được gán cho các ứng dụng. Để gán nhãn cuối cùng, thao tác ''OR'' được áp dụng cho đầu ra của cả hai nhóm. Do đó, nếu một trong hai nhóm gán nhãn RW cho ứng dụng, nó được phân loại là ransomware. Ứng dụng này chỉ được phân loại là không ransomware nếu cả hai mô hình của cụm tĩnh và động gán cho nó một nhãn tương tự (tức là NRW). Vì học có giám sát được sử dụng, bộ đào tạo bao gồm các mẫu ứng dụng Android được gán cho một trong hai lớp: Ransomware hoặc Non-Ransomware. Đối với các mô hình hòa tấu, chúng tôi sử dụng Naïve Bayes, Decision Tree (j48 / c4.5), Random Tree, Random Forest, Support Vector Classifier, Logistic Regression, Adaptive Boosting (Ada boosting), Gradient Boosting, Support Vector Machine with Sequential Minimal Optimization, JRip, v.v.

**Naïve Bayes** (NB) là một phân loại xác suất. Nó áp dụng lý thuyết xác suất và định lý Bayes để đưa ra giả định rằng các tính năng là độc lập [22]. Cây quyết định tạo ra các quy tắc để dự đoán các biến mục tiêu [22]. Thuật toán phân loại cây giúp dễ dàng hiểu được sự phân phối dữ liệu mong muốn. J48 (tức là triển khai Java mã nguồn mở C4.5 trong công cụ khai thác dữ liệu WEKA) thực hiện tính toán giá trị bị thiếu, cắt tỉa cây quyết định, phạm vi giá trị thuộc tính liên tục, dẫn xuất quy tắc, v.v. [23]. Cây ngẫu nhiên là một cây được tạo ngẫu nhiên từ một tập hợp các cây tiềm năng có một số k tính năng ngẫu nhiên tại mỗi nút. Trong trường hợp này, ''ngẫu nhiên'' có nghĩa là mỗi cây riêng lẻ có một nét tương đương trong tập hợp các cây mà từ đó một mẫu được rút ra. Sau đó, mỗi cây có thể nói là có một sự lan tỏa "đồng đều". Cây ngẫu nhiên có thể được tạo ra tốt, và việc tích hợp các bộ sưu tập lớn các cây ngẫu nhiên đòi hỏi các mô hình chính xác [22]. Một khu rừng ngẫu nhiên (RF) bao gồm một số cây quyết định ngẫu nhiên. Có hai loại ngẫu nhiên được xây dựng trong cây. Đầu tiên, mỗi cây được xây dựng trên một mẫu ngẫu nhiên từ dữ liệu gốc. Thứ hai, tại mỗi nút, một tập hợp con các tính năng được chọn ngẫu nhiên để tạo ra sự phân tách tốt nhất [22]. Mục tiêu của máy vector hỗ trợ (SVM) là phù hợp với dữ liệu được cung cấp và xác định siêu mặt phẳng phù hợp nhất để phân loại dữ liệu được cung cấp. Khi siêu mặt phẳng được xác định, các tính năng có thể được nhập vào bộ phân loại để xác định lớp dự đoán. Trong trường hợp có nhiều lớp, SVM sử dụng chiến lược "một đấu một".

**Hồi quy logistic** (LR) là một bộ phân loại tuyến tính tính toán xác suất hạn chế của kết quả và chọn xác suất có xác suất cao nhất. Tăng cường là một phương pháp tổng hợp phổ biến tạo ra một bộ phân loại mạnh từ các phân loại yếu khác nhau. Điều này được thực hiện bằng cách xây dựng một mô hình từ dữ liệu đào tạo và sau đó tạo một mô hình để sửa các lỗi của mô hình chính. Các mô hình được thêm vào cho đến khi bộ đào tạo được dự đoán hoàn hảo mà không có lỗi hoặc giá trị cực đoan. AdaBoost (AD) hoạt động bằng cách cân nhắc các quan sát. Trong quá trình này, các mẫu hoặc mẫu có vấn đề khó phân loại được cân nặng hơn và các mẫu được xử lý hiệu quả có trọng số yếu hơn. Chúng tôi đã sử dụng AdaBoost M1 với cơ sở SVM để đánh giá tổng hợp vì nó hoạt động tốt hơn AdaBoost với một loại người học yếu khác [22]. Gradient Boosting xác định điểm yếu bằng cách sử dụng gradient trong chức năng mất.

Đầu ra của những người học yếu khác nhau được kết hợp để chức năng mất của họ có thể được tối ưu hóa [22]. Hàm mất mát là thước đo mức độ phân loại dữ liệu cơ bản của mô hình dự đoán. Gradient boosting (GB) cho phép tối ưu hóa chức năng mất mát bằng cách thêm những người học yếu trong quá trình gradient descent. Support Vector Machine (SVM) kiểm tra, xác định và khớp các mẫu dữ liệu để phân loại. Nó sử dụng một siêu mặt phẳng để phân vùng dữ liệu thành các vùng của không gian n chiều. Siêu mặt phẳng giữ các giá trị của lề giữa các vùng ở mức tối đa. SVM sử dụng hàm kernel dẫn đến bề mặt phân loại phi tuyến thay vì siêu mặt phẳng tuyến tính. Tối ưu hóa tối thiểu tuần tự (SMO) là một thuật toán lặp lại để giải quyết các vấn đề tối ưu hóa phát sinh trong giai đoạn đào tạo của Máy vector hỗ trợ (SVM). SMO thực hiện phân mảnh vấn đề thành một chuỗi các vấn đề con nhỏ nhất có thể, sau đó được giải quyết bằng phân tích [24]. JRip là một thuật toán phân loại dựa trên quy tắc. Nó phát triển một trình học quy tắc tỷ lệ được gọi là '' Cắt tỉa gia tăng lặp đi lặp lại để tạo ra giảm lỗi (RIPPER)'' để trích xuất quy tắc trực tiếp từ dữ liệu và sử dụng các thuật toán bảo hiểm liên tiếp để tạo danh sách quy tắc được yêu cầu. Thuật toán trải qua bốn giai đoạn. (1) xây dựng quy tắc (2) cắt tỉa (3) tối ưu hóa và (4) lựa chọn [24].

**4. Experimental results**

Nghiên cứu này đã tiến hành đánh giá thực nghiệm để đánh giá hiệu quả của phương pháp Phân tích tập thể khác biệt lai được đề xuất của chúng tôi. Phương pháp đánh giá sử dụng kỹ thuật xác nhận chéo 10 lần để đánh giá mô hình được đề xuất. Trong xác thực chéo k-fold, tập dữ liệu ban đầu được chia ngẫu nhiên thành k tập con có kích thước bằng nhau. Một tập con duy nhất của các tập con k được sử dụng để xác nhận để kiểm tra thuật toán. Các tập con k − 1 còn lại được sử dụng làm tập huấn luyện. Quá trình xác thực chéo được lặp lại chính xác k lần (số lần gấp). Do đó, mỗi tập con k được sử dụng chính xác một lần làm tập xác thực [22]. Ưu điểm của chiến lược này là tất cả các quan sát được sử dụng cho cả đào tạo và thử nghiệm, và mỗi quan sát được sử dụng chính xác một lần để xác nhận. Sau các thí nghiệm k, trung bình có trọng số của độ chính xác phân loại được tính toán. Nghiên cứu này đã sử dụng các chỉ số hiệu suất được đưa ra trong Bảng 2, trong đó True Positive (TP) là kết quả dương tính chính xác giúp phát hiện ransomware khi có ransomware. True Negative (TN) là kết quả âm tính thực tế không phát hiện ransomware khi không có ransomware. Dương tính giả (FP) là kết quả dương tính giả, có nghĩa là ransomware được phát hiện khi không có ransomware. Âm tính giả (FN) là kết quả âm tính giả, có nghĩa là ransomware không được phát hiện khi có ransomware. Sức mạnh tính toán được sử dụng trong các thí nghiệm được liệt kê trong Bảng 3.

**4.1. Dataset**

Toàn bộ bộ dữ liệu được sử dụng trong nghiên cứu này bao gồm hai tập hợp con: phần mềm độc hại Android (Drebin4) và ứng dụng ransomware Android (RansomProber5) dưới dạng tệp .apk. Bộ dữ liệu phần mềm độc hại Drebin được sử dụng cho bộ dữ liệu phần mềm độc hại, một kho lưu trữ điểm chuẩn với hơn năm nghìn ứng dụng phần mềm độc hại (từ 179 họ phần mềm độc hại khác nhau). Bộ dữ liệu ransomware được sử dụng để phân tích đến từ kho lưu trữ ransomware được sử dụng cho các thí nghiệm trong nỗ lực nghiên cứu có tên RansomProber. Bộ dữ liệu RansomProber bao gồm hơn hai nghìn mẫu được lấy từ các cảnh báo bảo mật liên quan, báo cáo mối đe dọa từ các công ty chống vi-rút và blog bảo mật cite6chen2017uncovering. Bộ dữ liệu RansomProber cho thấy phạm vi bảo hiểm tốt của các gia đình ransomware Android hiện có. Bộ dữ liệu chứa 5500 ransomware và 2280 non-ransomware. Thử nghiệm đã sử dụng 275 mẫu của mỗi lớp cho các cuộc tấn công né tránh. Các bộ phân loại được sử dụng đã được đào tạo bằng cách sử dụng các đặc điểm hành vi của các ứng dụng ransomware và không ransomware và nhãn rõ ràng (tức là ransomware/non-ransomware). Phương pháp phát triển sử dụng các bộ dữ liệu rời rạc cho mục đích đào tạo và thử nghiệm. Nghiên cứu này đã sử dụng nhiều ứng dụng để đảm bảo rằng tập dữ liệu của chúng tôi không thiên vị. Bộ dữ liệu được sử dụng trong nghiên cứu này không giới hạn ở các ứng dụng có thuộc tính cụ thể có thể giúp tạo ra kết quả. Tuy nhiên, nếu các tính năng mới đến từ bất kỳ chức năng nào cho ứng dụng mở rộng, mô hình có thể kết hợp các tính năng được cập nhật. Bảng 2 mô tả các phương pháp được sử dụng để tính toán tất cả các giá trị. Các phép đo này được thực hiện bằng cách đánh giá từng thuật toán phân loại và các kết hợp khác nhau của việc học tập đồng bộ.

A screenshot of a document

Description automatically generated

**4.2. Static and dynamic feature selection using InfoGain**

Trong quá trình phát triển mô hình được đề xuất, nghiên cứu này đã thực hiện lựa chọn tính năng để chọn các thuộc tính quan trọng của tập dữ liệu phù hợp và hữu ích nhất để xác định các lớp ứng dụng (tức là RW / NRW). Để đạt được điều này, nghiên cứu thực hiện lựa chọn tính năng dựa trên tiêu chí thu được thông tin để tìm ra các tính năng phù hợp nhất bằng cách gán trọng số cho thông tin để làm nổi bật hiệu quả của các tính năng [22]. Phương pháp đã chọn 72 tính năng tĩnh từ 2911 tính năng sau khi áp dụng quy trình lựa chọn tính năng cho các tính năng tĩnh. Phương pháp này đã chọn 45 tính năng động từ 130 tính năng động. Trong công trình này, nghiên cứu kết hợp mô hình phân loại và đánh giá các kết hợp khác nhau như được đưa ra trong Bảng 4. Hình 2 mô tả mười tính năng hàng đầu thu được từ toàn bộ tập dữ liệu của các tính năng tĩnh và động bằng phương pháp InfoGain. Để kiểm tra hiệu suất của mô hình được đề xuất, nghiên cứu này đã sử dụng các biện pháp hiệu suất Độ chính xác, Thu hồi và Đo F. Bảng 2 mô tả các phương pháp được sử dụng để tính toán tất cả các giá trị. Các phép đo này được thực hiện bằng cách đánh giá từng thuật toán phân loại và các kết hợp khác nhau của việc học tập đồng bộ. Hình 3 và 4 cho thấy ảnh hưởng của việc xáo trộn tập dữ liệu đối với phép đo F trong quá trình phát triển mô hình và kết quả phân loại với một bộ phân loại và phân loại duy nhất với các nhóm khác nhau (đối với các tính năng tĩnh và động được chọn bằng phương pháp InfoGain). Các giá trị về độ chính xác và thu hồi nằm trong cùng một phạm vi do tập dữ liệu được sử dụng. Hình 3 cho thấy các chỉ số hiệu suất của mô hình được đề xuất của chúng tôi đối với dữ liệu tĩnh và động được xếp hạng bằng cách sử dụng các thuật toán học máy đơn lẻ. Ngược lại, Hình 4 hiển thị các chỉ số hiệu suất cho dữ liệu tĩnh và động được xếp hạng bằng cách sử dụng các thuật toán tổng hợp. Từ Hình 3 và 4, có thể thấy rằng các thuật toán tổng hợp hoạt động tốt so với các thuật toán đơn lẻ trên dữ liệu lai được xếp hạng. Trong số các thuật toán riêng lẻ, Ada Boost (AB) [22] là tốt nhất với các giá trị lần lượt là 0,853, 0,886 và 0,849 cho độ chính xác, thu hồi và đo F. Tuy nhiên, việc thực hiện kết hợp Random Tree + Random Forest + SVM với SMO là phân loại thành viên thấp hơn đáng kể so với tất cả các loại khác với độ chính xác 0,863, thu hồi 0,892 và 0,86 đo F.

A screenshot of a graph

Description automatically generated

A graph of different sizes and colors

Description automatically generated with medium confidence

A graph of different sizes and colors

Description automatically generated with medium confidence

Nghiên cứu đã thực hiện lựa chọn tính năng bằng cách sử dụng Phân tích thành phần chính (PCA). PCA là một thuật toán giảm kích thước cho phép chúng tôi xác định các mối tương quan và mẫu trong tập dữ liệu [25]. Do đó, tập dữ liệu có thể được chuyển đổi thành tập dữ liệu chiều thấp bằng cách loại bỏ các mối tương quan này mà không làm mất thông tin quan trọng. PCA là một thủ tục toán học chuyển đổi nhiều biến tương quan thành một vài biến không tương quan được gọi là các thành phần chính. Tập hợp con nhỏ các biến không tương quan này dễ làm việc hơn nhiều để xác định và sử dụng trong phân tích so với tập hợp lớn các tính năng tương quan. Hình 5 và 6 mô tả kết quả hoạt động của các bộ phân loại và tập hợp đơn lẻ trên dữ liệu tĩnh và động được chọn của PCA. Hình 5 minh họa rằng hiệu suất của LR cao nhất với độ chính xác, thu hồi và đo F 0,993. Hiệu suất SVM vẫn thấp nhất ở độ chính xác 0,481, thu hồi 0,231 và 0,312 đo F, không như mong đợi. Tất cả các bộ phân loại riêng lẻ khác ngoại trừ SVM hoạt động tốt đáng kể trên dữ liệu lai đã chọn. Hình 6 cho thấy rằng tập hợp với các bộ phân loại cơ sở ''SVM với SMO + hồi quy logistic + hồi quy logistic đơn giản + AdaBoostM1 với cơ sở SVM + Adaboosting'' vượt trội hơn tất cả các nhóm khác trên dữ liệu lai được chọn bởi PCA, đạt được độ chính xác, thu hồi và các biện pháp F 0.989. Màn trình diễn của các nhóm nhạc khác cũng rất đáng kể. Từ Hình 5 và 6, chúng ta có thể thấy rằng cả bộ phân loại riêng lẻ và bộ phân loại tổng hợp đều hoạt động tốt hơn nhiều trên dữ liệu được chọn bởi PCA so với dữ liệu được chọn bằng cách truy xuất thông tin. Loại bỏ các tính năng dư thừa làm tăng độ chính xác phân loại và giảm chi phí tính toán.

**4.4. Evaluation without feature selection experiments**

Hình 7 cho thấy hiệu suất của tất cả các bộ phân loại riêng lẻ trên dữ liệu lai bao gồm cả tính năng phân tích tĩnh và động (sử dụng một bộ phân loại duy nhất thay vì một tập hợp). Kết quả cho thấy hiệu suất của Ada Boost (AB) cao đáng kể trong số tất cả các phân loại đơn lẻ khác. Tất cả các thuật toán đơn lẻ đều hoạt động tốt cho dữ liệu lai được cung cấp, ngoại trừ SVM, có hiệu suất luôn thấp nhất trong tất cả các thử nghiệm. Hình 7 cho thấy hiệu suất của phương pháp được đề xuất của chúng tôi, một phân tích tổng hợp lai được giải thích trong Phần 3. Nó bao gồm hai mô hình học máy tập hợp khác nhau cho các bộ dữ liệu tính năng tĩnh và động. Kết quả của tất cả các nhóm khác nhau được thể hiện trong Hình 7. Họ chỉ ra rằng sự kết hợp với SVM với SMO + Hồi quy Logistic + Hồi quy Logistic đơn giản + AdaBoostM1 với cơ sở SVM và Ada Boosting vượt trội hơn tất cả các thuật toán học máy riêng lẻ và đồng bộ khác bằng cách đạt được độ chính xác, thu hồi và giá trị F 0,99. Độ chính xác của nhóm sử dụng SVM với SMO + Hồi quy hậu cần + AdaBoostM1 với thuật toán của thành viên cơ sở SVM và tập hợp sử dụng SVM với SMO + Hồi quy hậu cần + Thuật toán dựa trên rừng ngẫu nhiên là cao thứ hai trong số tất cả các nhóm khác với độ chính xác, thu hồi và đo F 0,997. Rõ ràng từ Hình 7 rằng hiệu suất cao nhất đạt được khi phân loại được thực hiện bằng cách sử dụng một tập hợp với các phân loại cơ sở:

A graph of different sizes and colors

Description automatically generated with medium confidence

1. SVM với SMO, hồi quy logistic, hồi quy Logistic đơn giản, AdaBoost với cơ sở SVM và Adaboosting;

2. SVM với SMO, hồi quy logistic, AdaBoost với cơ sở SVM;

3. SVM với SMO, hồi quy logistic và rừng ngẫu nhiên.

Nhóm đầu tiên đạt được hiệu suất cao nhất trong phân loại trên một tập dữ liệu lai nhất định, chỉ được cải thiện 0,3% so với hai nhóm còn lại: SVM với SMO + Hồi quy hậu cần + AdaBoostM1 với cơ sở SVM và SVM với SMO + Hồi quy hậu cần + Rừng ngẫu nhiên. Điều này cho thấy các nhóm này cung cấp kết quả tương tự trong phân loại và phát hiện ransomware Android.

**4.5. Adversarial evasion attacks experimentation**

Để xác nhận khả năng phục hồi của mô hình hòa tấu khác biệt lai được đề xuất chống lại các cuộc tấn công trốn tránh đối thủ, chúng tôi đã thử mô hình đồng bộ sử dụng các đầu vào được chế tạo. Các đầu vào được chế tạo này (để bắt chước một cuộc tấn công trốn tránh đối thủ) được tạo ra bởi những thay đổi nhỏ trong các vectơ tính năng ransomware Android đã biết. Chúng tôi đã đánh giá hiệu suất của mô hình được đề xuất trong việc giảm thiểu các cuộc tấn công vượt kiểm duyệt bằng cách thực hiện các thay đổi 1 bit, 10 bit và 20 bit đối với các vectơ tính năng đầu vào của ransomware đã biết. Hầu hết các kỹ thuật phát hiện ransomware Android dựa trên máy học hiện tại đều dễ bị tấn công trốn tránh đối thủ. Sự thay đổi bit trong vectơ tính năng đầu vào làm cho mô hình phân loại cơ bản bị bỏ qua. Để kiểm tra mô hình được đề xuất chống lại sự thay đổi như vậy trong vectơ tính năng đầu vào, chúng tôi đã chọn ngẫu nhiên 100 vectơ tính năng của ransomware đã biết. Chúng tôi đã thay đổi quyền của từng vectơ một chút. Điều này là do quyền là tính năng dễ bị tấn công nhất để mã hóa hoặc đổi tên. Chúng tôi đã thử nghiệm máy phân tích đồng bộ lai được đề xuất của chúng tôi với phần mềm không ransomware và các vectơ tính năng được chế tạo bằng cách sử dụng các mô hình hòa tấu khác nhau. Kết quả thử nghiệm của các mô hình quần thể khác nhau ML về Độ chính xác, Thu hồi và F-Measure được thể hiện trong Bảng 5. Có thể thấy rằng mỗi nhóm tạo ra các giá trị Độ chính xác, Thu hồi và Đo lường F 0,98% cho dữ liệu một bit và có xu hướng phân loại sai khi số lượng mô hình đầu vào tăng lên. Để kiểm tra thêm khả năng phục hồi của mô hình, chúng tôi đã chế tạo dữ liệu cho 100 vectơ tính năng được chọn ngẫu nhiên của ransomware Android. Trong mỗi vectơ tính năng, 10 bit liên quan đến quyền được thay đổi. Một lần nữa, mô hình hòa tấu độc đáo lai được thử nghiệm trên các vectơ tính năng được sửa đổi này.

**5. Discussion**

Từ Bảng 6, có thể thấy rằng bộ phân loại thành viên đạt được các giá trị chính xác, thu hồi và đo F 0,97, 0,98 và 0,99 cho dữ liệu chế tạo 1, 10 và 20 bit tương ứng và các giá trị này là thấp nhất trong số tất cả. Tất cả các mô hình khác trong

A screenshot of a graph

Description automatically generated

nhóm ML đạt được độ chính xác, thu hồi và số đo F là 0,99%. Độ chính xác của các nhóm khác nhau đối với dữ liệu được chế tạo 10 bit có thể được thấy trong Bảng 6, cho thấy xu hướng tương tự về hiệu suất của các nhóm khác nhau.

Sau khi đánh giá mô hình cho các đầu vào được chế tạo 1 bit và 10 bit và đạt được độ chính xác cao cho tất cả các nhóm ngoại trừ một nhóm, chúng tôi kiểm tra mô hình với dữ liệu đầu vào được chế tạo 20 bit. Các vectơ đặc tính đầu vào bịa đặt được xác định bằng cách sử dụng vectơ đặc tính của ransomware đã thay đổi 20 bit ngẫu nhiên (liên quan đến quyền). Việc thay đổi bit trong tính năng này đã thay đổi gần như toàn bộ khía cạnh quyền. Dựa trên dữ liệu quyền được trích xuất, người ta thấy rằng hầu như không có ứng dụng ransomware nào có thể yêu cầu hơn 20 quyền riêng lẻ. Chúng tôi đã đánh giá mô hình phân tích tập hợp duy nhất lai được đề xuất của chúng tôi bằng cách sử dụng các vectơ đặc trưng được chế tạo này. Bảng 6 cho thấy kết quả thử nghiệm thu được đối với các nhóm khác nhau. Từ kết quả kiểm tra chế tạo 20 bit, rõ ràng việc chế tạo các quyền đáng kể này không ảnh hưởng đến hiệu suất của mô hình tổ hợp riêng biệt lai, ngoại trừ một tổ hợp có độ chính xác và F-Measure lần lượt là 0,9 và 0,9, cho chế tạo 40 bit (thấp nhất trong tất cả). Kết quả của ba thử nghiệm này (tức là dữ liệu bị thay đổi 1 bit, 10 bit, 30 bit và 40 bit) chứng minh rằng phương pháp phân tích tập hợp lai được đề xuất có thể giảm thiểu các cuộc tấn công lẩn tránh đối nghịch. Phương pháp được đề xuất có thể phát hiện phần mềm ransomware Android và các mẫu giả mạo với độ chính xác cao. Theo hiểu biết tốt nhất của chúng tôi, mô hình đề xuất của chúng tôi là công trình nghiên cứu đầu tiên tập trung vào việc giảm thiểu các cuộc tấn công trốn tránh đối thủ thông qua học tập tổng hợp. Nó phân tích hành vi của các mẫu ransomware Android để phát hiện bản chất độc hại của chúng, khác với các phần mềm độc hại Android khác. Không có nghiên cứu nào gần đây chia sẻ quyền, văn bản, tính năng dựa trên mạng, nhật ký cuộc gọi hệ thống, mức sử dụng CPU và mức sử dụng bộ nhớ cũng như sử dụng tính năng học tập tổng thể để phân biệt phần mềm ransomware Android với phần mềm độc hại Android khác. Hơn nữa, kết quả của chúng tôi xác nhận kết luận của chúng tôi bằng cách đào tạo xuất sắc bộ phân tích tổng hợp để phân loại phần mềm ransomware Android và giảm thiểu các cuộc tấn công lẩn tránh đối thủ.

**6. Conclusion and future work**

Trong bài viết này, một phương pháp tiếp cận tổng hợp dựa trên học máy đã được trình bày nhằm xem xét các tính năng ứng dụng khác nhau để phát hiện phần mềm tống tiền. Các trình học tập hợp tĩnh và động bao gồm một số lượng lẻ các bộ phân loại như C4.5, Rừng ngẫu nhiên, JRip, Hồi quy logistic, SVM và AdaBoost. Mô hình tập hợp được huấn luyện ngoại tuyến và thử nghiệm trực tuyến để phân tích hành vi động. Trình phân loại meta dựa trên biểu quyết đa số đã đưa ra dự đoán cuối cùng. Nghiên cứu đã chỉ ra rằng quyền và nhật ký cuộc gọi hệ thống là hai tính năng quan trọng nhất để phát hiện và phân loại ransomware Android và non-ransomware.

Chúng tôi cũng điều tra khả năng của mô hình được đề xuất trong việc giảm thiểu các cuộc tấn công lẩn tránh đối thủ bằng cách thử nghiệm nó với các đầu vào giả trong thử nghiệm. Các kết quả hỗ trợ các quyết định của chúng tôi về việc đào tạo các bộ phân tích tổng hợp để phát hiện phần mềm tống tiền và giảm thiểu các cuộc tấn công lẩn tránh, vì chúng tôi thu được kết quả tốt. Cơ chế phân tích tập hợp riêng biệt được đề xuất cho thấy kết quả đầy hứa hẹn bằng cách đạt được độ chính xác, thu hồi và thước đo F cao trong phát hiện phần mềm tống tiền trên Android. Mô hình đề xuất chứng tỏ là một mô hình có khả năng phục hồi chống lại các cuộc tấn công lẩn tránh đối thủ bằng cách đạt được độ chính xác tốt trên các đầu vào được chế tạo 1 bit, 10 bit, 20 bit, 30 bit và 40 bit được cung cấp.

Vấn đề nghiên cứu trong tương lai cần giải quyết các đặc điểm của các trường hợp ransomware độc hại, các cuộc tấn công hiệu quả, trích xuất tính năng mạnh mẽ và hiệu quả về chi phí, ước tính tính năng độc hại, số liệu để xác thực hiệu suất phòng thủ độc hại và các biện pháp đối phó được thiết kế để bảo vệ ransomware. Chúng tôi cũng có kế hoạch phân tích các sự kiện tuần tự và tác động của chúng lên các mẫu của kẻ tấn công trong

tương lai.

**Tuyên bố về lợi ích cạnh tranh**

Các tác giả tuyên bố rằng họ không có lợi ích tài chính hoặc mối quan hệ cá nhân cạnh tranh nào có thể ảnh hưởng đến công việc được báo cáo trong bài viết này.