Journal of Modeling in Engineering 23 (2025) 47-57

Semnan University

Journal of Modeling in Engineering

Journal homepage: https://modelling.semnan.ac.ir/

ISSN: 2783-2538

Research Article

Malware Detection in Android Operating System using Convolutional Neural Network and Long Short-Term Memory Network

Aliakbar Tajari Siahmarzkooh [a,*], Ali Rahimi Hosseiabad [b]

[a] *Department of Computer Sciences, Faculty of Sciences, Golestan University, Gorgan, Iran*
[b] *Department of Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran*

PAPER INFO

Paper history:

Received: 2023-09-29
Revised: 2024-06-08
Accepted: 2024-06-11

Keywords:

Malware detection;
Convolutional Neural Network (CNN);
Long Short-Term Memory (LSTM);
MalMemAnalysis-2022 dataset.

ABSTRACT

The use of mobile phones with Android operating system is expanding day by day. Android itself does not have a powerful malware detection tool. Therefore, attackers easily enter people's privacy through their mobile phones and put them at serious risk. So far, a lot of research has been done on malware detection. One of the main problems of these solutions is the low accuracy in multi-class detection on the dataset or the failure to achieve the desired result in both types of binary and multi-class detection. In this paper, by using Convolutional Neural Network (CNN) and changing the number of different layers, we have tried to extract the maximum number of important features from the dataset. In the data classification phase, we use the Deep Learning-based algorithm named Long Short-Term Memory (LSTM) to classify the data with the maximum possible accuracy by testing it on the selected features. The test results on the new MalMemAnalysis-2022 dataset show that the use of these two algorithms and the change in the number of layers can lead to 99.99% and 99.71% accuracies in binary and multi-class classification in malware detection, respectively, which is superior to existing methods.

DOI: https://doi.org/10.22075/jme.2024.31932.2537

*** Corresponding Author.**
E-mail address: a.tajari@gu.ac.ir

How to cite this article:
Tajari Siahmarzkooh, A. and Rahimi HosseinAbad, A. (2025). Malware Detection in Android Operating System using Convolutional Neural Network and Long Short-Term Memory Network. Journal of Modeling in Engineering, 23(80), 47-57. doi: 10.22075/jme.2024.31932.2537

مقاله پژوهشی

تشخیص بدافزار در سیستم‌عامل اندروید با استفاده از شبکه عصبی کانولوشن و شبکه حافظه طولانی کوتاه‌مدت

علی‌اکبر تجری سیاه‌مرزکوه [۱،*] ، علی رحیمی حسین‌آباد [۲]

اطلاعات مقاله

دریافت مقاله: ۱۴۰۲/۰۷/۰۷
بازنگری مقاله: ۱۴۰۳/۰۳/۰۹
پذیرش مقاله: ۱۴۰۳/۰۳/۲۲

واژگان کلیدی:
تشخیص بدافزار،
شبکه عصبی کانولوشن (CNN)،
شبکه حافظه طولانی کوتاه-مدت (LSTM)،
مجموعه داده MalMemAnalysis-2022

چکیده

استفاده از تلفن‌های همراه با سیستم‌عامل اندروید روز به‌روز در حال گسترش است. سیستم‌عامل اندروید به‌خودی خود ابزار قدرتمندی برای تشخیص بدافزار ندارد. از این‌رو، مهاجمان به‌راحتی از طریق گوشی تلفن همراه افراد وارد حریم خصوصی آنها شده و آنها را در معرض خطر جدی قرار می‌دهند. تاکنون تحقیقات زیادی بر روی تشخیص بدافزار صورت گرفته است. یکی از مشکلات عمده این راهکارها، دقت پایین در تشخیص چند کلاسه روی مجموعه داده‌ها و یا عدم حصول نتیجه مطلوب در هر دو نوع تشخیص دودویی و چند کلاسه است. در این مقاله با استفاده از شبکه عصبی کانولوشن (CNN) و تغییر در تعداد لایه‌های مختلف، سعی کرده‌ایم تا حداکثر تعداد ویژگی‌های مهم را از مجموعه داده استخراج نماییم. در فاز طبقه‌بندی داده‌ها نیز از الگوریتم یادگیری شبکه حافظه طولانی کوتاه‌مدت (LSTM) استفاده می‌کنیم تا با آزمایش آن بر روی ویژگی‌های انتخاب شده، داده‌ها با حداکثر دقت ممکن طبقه‌بندی شوند. نتایج آزمایش بر روی مجموعه داده جدید MalMemAnalysis-2022 نشان می‌دهد که استفاده از این دو الگوریتم و تغییر در تعداد لایه‌ها می‌تواند در بهترین حالت به‌ترتیب منجر به دقت‌های ۹۹.۹۹٪ و ۹۹.۷۱٪ در دسته‌بندی دودویی و چند کلاسه در تشخیص بدافزار شود که نسبت به روش‌های موجود برتری دارد.

DOI: https://doi.org/10.22075/jme.2024.31932.2537

۱- مقدمه

استفاده فراگیر از تلفن‌های همراه، آنها را به بخشی ضروری از فعالیت‌های روزمره زندگی تبدیل کرده است [۱]. اندروید بدون شک نسبت به سایر سیستم‌عامل‌های موجود محبوبیت بیشتری در بازار به‌دست آورده است. از طرفی، استفاده فراگیر از اندروید توجه بیشتر مهاجمان سایبری را نیز به‌خود جلب کرده است. نقص‌های امنیتی در گوشی‌های هوشمند اندرویدی نشان می‌دهد که آنها در برابر حملات فیشینگ پیامکی بسیار آسیب‌پذیر هستند. معماری باز سیستم‌عامل اندروید نیز، اثرات مخرب جرایم سایبری بر پلتفرم اندروید را تشدید می‌کند [۲].

نگرانی‌های اصلی امنیت سایبری شامل حملات بدافزار، عدم سرویس‌دهی، باتنت‌ها، روت‌کیت‌ها، نفوذها و باج‌افزارها است. بدافزار نرم‌افزاری است که شامل ویروس‌ها، ابزارهای

* پست الکترونیک نویسنده مسئول: a.tajari@gu.ac.ir
۱. دانشکده علوم، گروه علوم کامپیوتر، دانشگاه گلستان، گرگان، ایران
۲. دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه فنی و حرفه‌ای، تهران، ایران

استناد به این مقاله:
تجری سیاه مرزکوه، علی اکبر و رحیمی حسین آباد، علی . (۱۴۰۴). تشخیص بدافزار در سیستم عامل اندروید با استفاده از شبکه عصبی کانولوشن و شبکه حافظه طولانی کوتاه‌مدت. مدل سازی در مهندسی، ۲۳(۸۰)، ۴۷-۵۷. doi: 10.22075/jme.2024.31932.2537

تبلیغاتی مزاحم، اسب‌های تروجان، نرم‌افزارهای جاسوسی و غیره است و می‌تواند به رایانه‌ها و دستگاه‌های وب آسیب برساند. در حمله بدافزار، مهاجم می‌تواند بدون هیچ‌گونه آگاهی به شبکه دسترسی پیدا کند و کنترل کامل را در دست بگیرد. حملات بدافزار به‌دلیل استفاده گسترده و اقدامات امنیتی ناکافی، تهدیدی بالقوه برای شبکه‌های تلفن همراه محسوب می‌شوند [۳]. بنابراین، لازم است راهکاری برای شناسایی حملات بدافزار وجود داشته باشد تا اقدامات فوری انجام شود و سیستم یا دستگاه قبل از به خطر افتادن، ایمن شود.

نفوذگران از فایل‌های apk برای سوءاستفاده از آسیب‌پذیری‌های شناخته شده و حمله سایبری به سیستم استفاده می‌کنند. از آنجایی‌که فایل‌های apk به‌راحتی در دسترس کاربران اندروید قرار می‌گیرند، به منبعی برای حملات تبدیل شده‌اند. از طرفی، اخلال‌گران از مجوزهایی که توسط خدمات گوگل‌پلی به برنامه‌ها داده شده، برای ورود غیر مجاز به سیستم استفاده می‌کنند. علاوه بر این، خدمات این سرویس، دسترسی به برخی از ویژگی‌ها را بدون رضایت کاربران فراهم می‌کند. با این حال، اگر کاربری بخواهد آنها را به‌صورت دستی غیر فعال کند، عملکرد سیستم دچار اختلال می‌شود. اعطای چنین مجوزهایی در زمان نصب، سیستم را در برابر حملات آسیب پذیرتر می‌کند و حریم خصوصی کاربران را به‌خطر می‌اندازد و اگر هم کاربران، این مجوزها را رد کنند نمی‌توانند به‌درستی از برنامه استفاده نمایند [۴].

نفوذگران با ورود به سیستم، اقدام به سرقت داده‌ها نموده و به آنها آسیب وارد می‌کنند. این اقدام آنها از طریق دسترسی به برنامه‌هایی که مجوز دسترسی به آنها را ندارند و در نتیجه مشاهده داده‌های کاربر صورت می‌گیرد. با توجه به اینکه دسترسی به اجزای سخت‌افزاری از قبیل دوربین و میکروفون تنها در صورتی امکان‌پذیر است که برنامه مورد استفاده آنها مجوز استفاده از آن اجزا را داشته باشد، بنابراین، مجوزها نقشی حیاتی در میان سایر ویژگی‌ها در تشخیص بدافزار دارند.

در حالت کلی، تجزیه و تحلیل بدافزار به دو دسته ایستا و پویا تقسیم‌بندی می‌شود [۵]. تشخیص بدافزار به‌صورت ایستا روشی است که با روش‌های مبتنی بر امضا، مبتنی بر مجوز و مبتنی بر کد بایتی شناسایی می‌شود و بسیار ساده بوده و قابلیت شکستن دارد. از طرفی، روش پویا راهکاری برای تشخیص است که در آن برنامه‌ها در پلتفرم‌های مجزا اجرا می‌شوند و رفتارها در حین اجرای فایل مشکوک ردیابی می‌شوند. روش‌های سنتی در مواردی که یک بدافزار به‌طور گسترده رشد می‌کند ضعیف عمل می‌کنند. برای رفع این مشکل، روش‌های یادگیری ماشین برای شناسایی بدافزارها ارائه شده‌اند. با این حال، این روش‌ها به تلاش بیشتری برای استخراج ویژگی‌ها [۶] نیاز دارند. اخیراً، مدل‌های یادگیری عمیق و شبکه عصبی کانولوشن برای شناسایی بدافزار توسط محققان بسیاری مورد استفاده قرار گرفته‌اند.

ابزارهای تشخیص مختلفی برای محافظت از سیستم در مقابل حملات مخرب وجود دارد که برنامه‌های اندروید را تجزیه و تحلیل کرده و آنها را به‌عنوان مخرب طبقه‌بندی می‌کند.

با توجه به اینکه در رویکرد ایستا، تمام بخش‌های برنامه برای بررسی دقیق در دسترس هستند محققان تشخیص ایستا را بر پویا ترجیح می‌دهند. در مقابل، تشخیص پویا به عواملی مانند مقدار یک ورودی خاص برای وقوع یک رخداد خطرناک وابسته است. در این شرایط، رویکرد تشخیص ایستا تجزیه و تحلیل عمیق و دقت بهتری نسبت به تحلیل پویا ارائه می‌کند. همچنین، با توجه به افزایش قابل توجه داده‌های مخرب، تشخیص دستی نفوذها عملاً غیر ممکن است. علاوه بر این، بسیاری از روش‌های تشخیص بدافزار که محققان معرفی کرده‌اند به منابع زیادی نیاز دارند. هزینه و پیچیدگی زمان اجرای روش تشخیص در پایگاه داده بزرگ در این روش‌ها زیاد است. بنابراین، الگوریتم‌های یادگیری ماشین می‌توانند داده‌ها را با دقت در زمان کمتری طبقه‌بندی کنند [۷].

از آنجایی‌که تعداد ویژگی‌های موجود در فایل‌های اندرویدی بسیار زیاد است و سربار محاسباتی و از طرفی کاهش دقت سیستم تشخیص نفوذ را به‌دنبال دارد، منجر به زمان پردازش بالا و کاهش عملکرد کلی سیستم نیز می‌شود. محققان زیادی در حوزه تشخیص بدافزار کار کرده‌اند. با این حال، در زمینه استخراج ویژگی‌های مؤثر برای شناسایی بدافزار کار زیادی انجام نگرفته است. حذف و کنار گذاشتن ویژگی‌های نامرتبط و کم‌اثر از مجموعه ویژگی‌ها قبل از ورود آنها به الگوریتم یادگیری ماشین [۸] می‌تواند منجر به بهبود عمل پردازش و احتمالاً دقت تشخیص شود. در واقع، انتخاب ویژگی‌های مهم بخش ضروری از روش‌های تشخیص بدافزار است، زیرا به‌وضوح می‌تواند بر عملکرد

تشخیص نفوذ تأثیر مثبت بگذارد. هدف این مقاله استخراج مجموعه‌ای از ویژگی‌های مؤثر یا ویژگی‌های مرتبط است. مجموعه ویژگی‌های انتخاب شده می‌تواند تجزیه و تحلیل دقیق‌تری را ارائه دهد و منجر به ارائه یک مدل تشخیص بسیار دقیق و قابل اعتماد شود. در این مقاله، تلاش می‌کنیم تا مجموعه ویژگی‌های مرتبط را با استفاده از کاهش ویژگی‌ها ایجاد نموده و آن را برای عملیات طبقه‌بندی آماده کنیم.

در این مقاله، یک مدل انتخاب ویژگی پیشنهاد می‌شود که می‌تواند ویژگی‌های مرتبط را از یک مجموعه وسیع استخراج نماید. این مدل از دو بخش تشکیل شده است که بخش اول کاهش ویژگی‌ها و بخش دوم آزمایش بر روی ویژگی‌های انتخاب شده است.

در روش پیشنهادی، ابتدا ویژگی‌ها به‌عنوان ورودی دریافت شده و سپس به یک فاز کاهش ویژگی ارسال می‌شوند. در گام اول، ویژگی‌هایی که از لحاظ رفتاری با یکدیگر وابستگی دارند شناسایی شده و حذف می‌شوند. همچنین برخی از ویژگی‌ها در طبقه‌بندی مفید نیستند، زیرا وابستگی کمی با مقادیر ویژگی کلاس دارند، از این‌رو این دسته نیز حذف می‌شوند.

در گام بعدی، با استفاده از شبکه حافظه طولانی کوتاه‌مدت (LSTM) که یک معماری شبکه عصبی بازگشتی است اقدام به طبقه‌بندی داده‌ها می‌کنیم. این ابزار قدرتمند مبتنی بر یادگیری عمیق، بر خلاف شبکه عصبی سنتی دارای اتصالات بازخوردی است که به آن اجازه می‌دهد تا کل توالی داده‌ها را پردازش کند.

در نهایت، برخی معیارهای ارزیابی عملکرد مانند دقت، یادآوری، صحت، امتیاز F1 و چند پارامتر مفید دیگر محاسبه شده و بر اساس آن، کارآیی روش پیشنهادی بررسی می‌شود [۹].

ادامه مقاله به شرح زیر است: بخش ۲ کارهای مرتبط را مورد بحث قرار می‌دهد. بخش ۳ رویکرد پیشنهادی را نشان می‌دهد و بخش ۴ به بررسی معیارهای ارزیابی و نتایج و تجزیه و تحلیل می‌پردازد. بخش ۵ تجزیه و تحلیل با رویکرد مقایسه‌ای از نتایج را نشان می‌دهد.

۲- پیشینه تحقیق

در این بخش، برخی از کارهای پیشین انجام شده در حوزه تشخیص حملات اندرویدی را بررسی می‌کنیم.

لی و همکاران [۱۰]، یک ابزار تشخیص بدافزار مبتنی بر تجزیه و تحلیل مجوزها را ارائه نمودند. با بررسی‌هایی که در کار تحقیقاتی آنها صورت گرفت، طبقه‌بندی‌کننده ماشین بردار پشتیبان به مقادیر دقت، یادآوری، صحت و امتیاز F1 بالاتر از ۹۰ درصد دست یافته است.

ژائو [۱۱]، یک سیستم تشخیص بدافزار مبتنی بر رفتار برای اندروید ارائه کرد. در روش پیشنهادی او، فراخوانی API برنامه‌های متعدد به‌عنوان بردار ویژه در طول فرآیند آموزش با استفاده از تحلیل معکوس بازیابی می‌شوند. نتایج آزمایش نشان می‌دهد که مقادیر پارامتر دقت و صحت این روش به ۹۳ درصد می‌رسد.

وانگ و همکاران [۱۲]، یک روش ترکیبی مبتنی بر شبکه عصبی کانولوشن (CNN) و رمزگذار خودکار عمیق (DAE) ارائه کردند. برای اطمینان از دقت بالاتر هنگام شناسایی نرم‌افزارهای مخرب، چندین CNN برای جستجوی برنامه‌های مخرب اندروید و بازسازی ویژگی‌ها استفاده شده است. این روش در مقایسه با روش‌های یادگیری ماشین سنتی، بهبود قابل‌توجهی در شناسایی نرم‌افزارهای مخرب در اندروید نشان می‌دهد. به‌طور مشخص، کارآیی این مدل ۵ درصد بهتر از ماشین بردار پشتیبان است. کیم و همکاران [۱۳]، یک معماری منحصر به‌فرد برای شناسایی نرم‌افزارهای مخرب در اندروید پیشنهاد کردند. به‌منظور اصلاح ویژگی‌ها و به‌دست آوردن ویژگی‌های مؤثرتر برای تشخیص بدافزار، یک رویکرد استخراج ویژگی مبتنی بر شباهت ارائه شده است. چارچوب پیشنهادی از ویژگی‌های مختلف برای به تصویر کشیدن ویژگی‌های برنامه‌های اندروید از دیدگاه‌های مختلف استفاده می‌کند. چندین مرحله آموزش بر روی نمونه داده‌ها به‌منظور ارزیابی عملکرد فاز آموزش صورت گرفته است.

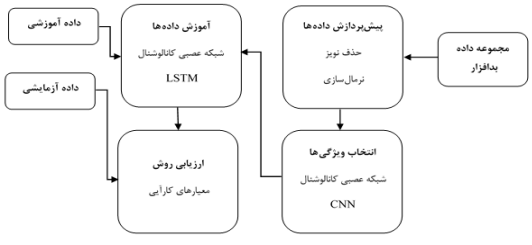
ژو و همکاران [۱۴]، روش یادگیری ویژگی بدون نظارت را معرفی کردند که برای آموزش ویژگی‌های مؤثر به‌عنوان اولین گام در چارچوب تشخیص ویروس استفاده می‌شود. آنها یک الگوریتم یادگیری شبکه عمیق ترکیبی از MSAE و SDAE به نام SHLMD با گسترش روش یادگیری عمیق رمزگذار خودکار به نام Stacked Denoising ایجاد کردند. به‌منظور بهبود قابلیت‌های تشخیص بدافزار، ویژگی‌های مهم‌تر و مؤثرتر استخراج شده و برای آموزش مدل تشخیص بدافزار از روش‌های طبقه‌بندی مانند ماشین بردار پشتیبان یا k- نزدیکترین همسایگی استفاده شده است.

هاسگاوا و همکاران [۱۵]، روشی سریع و آسان برای شناسایی بدافزار اندروید پیشنهاد کردند. فایل خام برنامه اندروید با استفاده از بخش بسیار کوچکی از یک شبکه عصبی کانولوشن یک بعدی ارزیابی می‌شود. روش پیشنهادی آنها با استفاده از اعتبارسنجی k-fold Cross Validation با مقدار k=10 قادر به تشخیص بدافزار با دقت حدود ۹۷٪.۰۴ بوده است.

فنگ و همکاران [۱۶]، شبکه‌های عصبی عمیق خاصی به نام Android MobiTive را برای تشخیص بدافزار پیشنهاد کردند که یک محیط تشخیص دستگاه تلفن همراه را ارائه می‌دهد که به‌صورت بلادرنگ عمل می‌کند. روش آنها از استخراج ویژگی مبتنی بر کد منبع و دسته‌بندی بر پایه یادگیری عمیق استفاده می‌کند که منجر به دقت تشخیص بالا می‌شود.

۳- روش پیشنهادی

در این بخش در مورد روش پیشنهادی برای تشخیص بدافزار اندروید به‌همراه طرح کلی از نحوه پیاده‌سازی آن شرح می‌دهیم. در این مقاله، یک راهکار تشخیص بدافزار اندروید پیشنهاد شده است که با حذف ویژگی‌های بی‌فایده با استفاده از الگوریتم انتخاب ویژگی مبتنی بر شبکه عصبی کانولوشن [۱۷ و ۱۸] و در ادامه آموزش ویژگی‌های مؤثر با استفاده از الگوریتم یادگیری عمیق LSTM [۱۹ و ۲۰]، عملیات دسته‌بندی داده‌ها انجام می‌شود. چهار بخش اساسی این روش عبارتند از: ۱- پیش‌پردازش اولیه داده‌ها شامل حذف ویژگی‌های وابسته و نرمال‌سازی داده‌ها، ۲- استفاده از الگوریتم CNN برای شناسایی ویژگی‌های مؤثر و حذف ویژگی‌های بی‌فایده از مجموعه داده‌ها جهت آماده سازی داده‌ها برای عملیات آموزش و آزمایش داده‌ها، ۳- استفاده از الگوریتم یادگیری عمیق مبتنی بر LSTM برای دسته‌بندی داده‌ها و ۴- ارزیابی روش ارائه شده بر اساس معیارهای کارآیی.



شکل ۱- فلوچارت کلی روش پیشنهادی

شکل (۱)، فلوچارت کلی روش پیشنهادی را نشان می‌دهد. همان‌طور که در این شکل مشاهده می‌شود روش پیشنهادی از چهار مرحله اساسی برای تشخیص بدافزار تشکیل شده است که در مورد هر یک از این مراحل در ادامه توضیح داده خواهد شد.

۳-۱- پیش‌پردازش داده‌ها

در این مرحله، داده‌های خام به‌عنوان ورودی دریافت شده و عملیات زیر به‌ترتیب روی آن صورت می‌گیرد:

۳-۱-۱- تحلیل همبستگی

با محاسبه ضریب همبستگی بین ویژگی‌ها که از رابطه ۱ به‌دست می‌آید، آن دسته از ویژگی‌هایی که وابستگی مثبت دارند کنار گذاشته می‌شوند تا از حجم داده‌های مجموعه داده کاسته شود.

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B} \qquad (۱)$$

در این رابطه، n، $\bar{A}$، $\bar{B}$، σ_A و σ_B به‌ترتیب نشان دهنده تعداد نمونه‌ها، میانگین مقادیر ویژگی A، میانگین مقادیر ویژگی B، انحراف معیار استاندارد A و انحراف معیار استاندارد B هستند.

برای مقدار $r_{A,B}$ سه حالت پیش می‌آید. اگر مقدار بزرگتر از صفر باشد نشان دهنده وابستگی مثبت بین A و B، اگر کوچکتر از صفر باشد بیانگر وابستگی منفی بین آن دو و در صورتی‌که مقدار آن صفر باشد نشان دهنده مستقل بودن A و B است. در دو حالت اول به‌دلیل وجود وابستگی، یکی از ویژگی‌ها حذف می‌شود.

۳-۱-۲- نرمال‌سازی داده‌ها

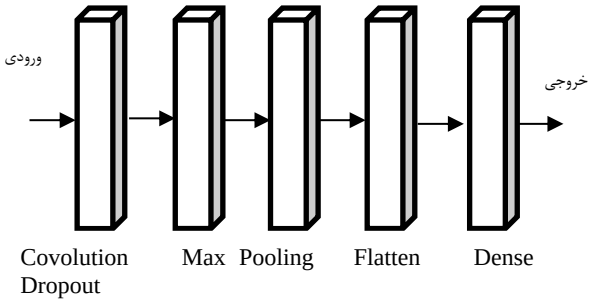
نرمال‌سازی داده‌ها روشی است که در داده‌کاوی برای تبدیل مقادیر یک مجموعه داده به یک مقیاس مشترک استفاده می‌شود [۲۱]. این موضوع قبل از پردازش نهایی داده‌ها الزامی است، زیرا بسیاری از الگوریتم‌های یادگیری ماشین به مقیاس ویژگی‌های ورودی حساس هستند و زمانی‌که داده‌ها نرمال‌سازی شوند، می‌توانند نتایج بهتری تولید کنند. در اینجا از نرمال‌سازی Z-Score استفاده می‌شود که در آن، مقادیر یک ویژگی برای رسیدن به میانگین با مقدار صفر و انحراف معیار استاندارد یک مقیاس می‌شوند. این کار با کم کردن میانگین ویژگی از هر مقدار و سپس تقسیم بر انحراف استاندارد (رابطه ۲) انجام می‌شود.

$$v' = \frac{v - \mu_A}{\sigma_A} \qquad (۲)$$

۳-۲- انتخاب ویژگی‌ها

با هدف به حداقل رساندن پیچیدگی مدل تشخیص حملات

بدافزار و حفظ دقت تشخیص بالا، کاهش ویژگی در مدل تشخیص حمله پیشنهادی را بر اساس معماری CNN طراحی کرده‌ایم که شامل لایه‌های کانولوشن، حداکثر جمع‌آوری (max-pooling) مسطح، متراکم و لایه‌های کاملاً متصل است. شکل (۲)، شبکه کانولوشن پیشنهادی برای انتخاب ویژگی‌های مؤثر را نشان می‌دهد.



شکل ۲- نمایش شبکه عصبی کانولوشن برای استخراج ویژگی‌ها

CNN یک روش یادگیری عمیق است که از لایه‌های مختلفی مانند کانولوشن، لایه‌های ادغام و لایه‌های کاملاً متصل تشکیل شده است. CNN معمولاً برای طبقه‌بندی تصویر و تشخیص صدا استفاده می‌شود. در این مقاله از CNN برای آماده‌سازی داده‌ها جهت شناسایی فعالیت‌های مخرب در اندروید استفاده می‌کنیم.

در بخش نتایج آزمایش با تغییر تعداد هر سطح از لایه‌ها اقدام به بررسی میزان دقت مدل می‌کنیم. به‌طور مثال، شبکه‌ای را فرض کنید که شامل دو لایه کانولوشن، یک لایه حداکثر تجمع، یک لایه مسطح، دو لایه متراکم و یک لایه کاملاً متصل است. ورودی آن در اولین لایه شامل مقدار (۱۸ و unknown) می‌باشد که مقدار اول، تعداد پویای نمونه‌ها و مقدار دوم، تعداد ویژگی‌های ورودی است. ۴۰ فیلتر در دو لایه اول کانولوشن استفاده شده است که خروجی به شکل (۱۶ و unknown) تولید می‌کند. خروجی لایه‌های کانولوشن به‌عنوان ورودی به لایه max-pooling وارد می‌شود. خروجی تولید شده توسط این لایه (۱۴ و unknown) است. در نهایت پس از گذر از آخرین لایه کاملاً متصل، خروجی (۱۰ و unknown) خواهد بود. عبور از این لایه‌ها نه تنها مهم‌ترین ویژگی‌ها را استخراج می‌کند بلکه نویزهای احتمالی را نیز کاهش می‌دهد.

$$x_k = b_k + \sum_{i=1}^{N}(s_i, w_{ik}) \quad (۳)$$

$$y_k = f(x_k) \quad (۴)$$

رابطه ۳ و ۴، عملکرد لایه کانولوشن را نشان می‌دهد که در آن x_k ورودی لایه است.

خروجی لایه قبل با s_i نشان داده می‌شود و w_{ik} نشان دهنده هسته از i تا k است. b_k مقدار بایاس نورون در لایه کانولوشن است. تابع فعال‌سازی ReLU با $f()$ نشان داده می‌شود. رابطه ۵، ReLU را توصیف می‌کند. y_k خروجی لایه کانولوشن بعدی است.

$$f(x_k) = \max(0, x_k) \quad (۵)$$

۳-۳- طبقه‌بندی داده‌ها

LSTM (حافظه طولانی کوتاه‌مدت) یک معماری شبکه عصبی بازگشتی (RNN) است که به‌طور گسترده در یادگیری عمیق استفاده می شود. این روش برای یافتن وابستگی‌های بلندمدت، عالی و برای کارهای پیش‌بینی توالی ایده‌آل است. برخلاف شبکه‌های عصبی سنتی، LSTM دارای اتصالات بازخوردی است که به آن اجازه می‌دهد تا کل توالی داده‌ها و نه فقط نقاط داده‌ای جداگانه را پردازش کند. این امر آن را در درک و پیش‌بینی الگوها در داده‌های متوالی مانند سری‌های زمانی، متن و گفتار بسیار مؤثر می‌کند.

حافظه کوتاه‌مدت، مشکل گرادیان ناپدید شدن در RNN را حل می‌کند. در سطح بالا، LSTM بسیار شبیه یک سلول RNN عمل می‌کند. معماری شبکه LSTM از سه قسمت تشکیل شده است.

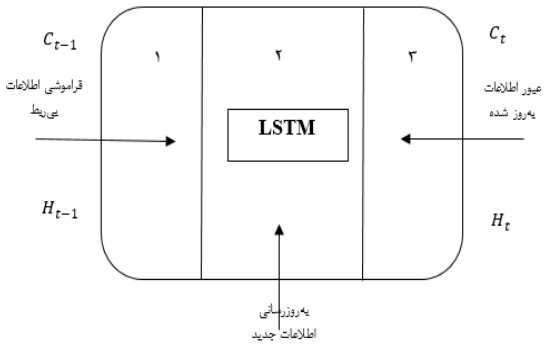
بخش اول انتخاب می‌کند که آیا اطلاعاتی که از مهر زمانی قبلی به‌دست می‌آید باید به‌خاطر سپرده شود یا نامربوط است و می‌تواند فراموش شود. تابع فعال‌سازی از طریق رابطه ۶ محاسبه می‌شود به‌گونه‌ای که در آن b بردار بایاس، h_{t-1} خروجی بلوک قبلی، x_t دنباله ورودی. W_{hf} و W_{xf} به‌ترتیب، ماتریس وزن بردارهای ورودی و خروجی از سلول‌های قبلی و فعلی را نشان می‌دهند. σ نیز توسط تابع سیگموئید ۷ ارائه می‌شود.

$$f_t = \sigma(W_{hf}h_{t-1} + W_{xf}x_t + b_f) \quad (۶)$$

$$\sigma(x) = (1 + e^{-x})^{-1} \quad (۷)$$

در قسمت دوم، سلول سعی می‌کند اطلاعات جدیدی را از ورودی این سلول بیاموزد. در نهایت، در بخش سوم، سلول اطلاعات به‌روز شده را از مهر زمانی فعلی به مهر زمانی بعدی منتقل می‌کند. این چرخه LSTM یک مرحله یکباره

در نظر گرفته می‌شود.
درست مانند یک RNN ساده، یک LSTM نیز دارای یک حالت پنهان است که در آن H(t-1) نشان دهنده وضعیت پنهان زمان قبلی و H_t وضعیت پنهان زمان فعلی است. علاوه بر آن، LSTM همچنین دارای یک حالت سلولی است که به ترتیب با C(t-1) و C(t) برای مهرهای زمانی قبلی و فعلی نشان داده می‌شود.
در اینجا حالت پنهان به‌عنوان حافظه کوتاه‌مدت و حالت سلولی به‌عنوان حافظه بلندمدت شناخته می‌شود (شکل ۳).



شکل ۳. سیستم عملکردی LSTM

در زیر چند نمونه از نحوه استفاده لایه ورودی از بردار ورودی فعلی برای تصمیم‌گیری اینکه چه داده‌هایی ممکن است در سلول حافظه ذخیره شوند، آورده شده است.
اینکه کدام خروجی ممکن است در مرحله زمانی فعلی ارسال شود توسط لایه خروجی تعیین می‌شود که در رابطه ۸ مشخص شده است:

$$O_t = \sigma\left(W_{ho}h_{t-1} + W_{xo}x_t + b_o\right) \qquad (۸)$$

وضعیت فعلی سلول با یکپارچه‌سازی به‌همراه ورودی محاسبه می‌شود:

$$C_t = f_t \odot C_{t-1} + i_t \odot C_t' \qquad (۹)$$

که در آن C_t شرایطی است که سلول در مرحله زمانی t و $\odot$ نشانگر ضرب عنصر بردار معروف به حاصلضرب هادامارد است. $f_t \odot C_{t-1}$ و $i_t \odot C_t'$ تعیین می‌کنند که چه محتوایی باید بین ورودی فعلی و حالت سلول قبلی منتقل شود. از این رو، C_t' مطابق با تابع فعال‌سازی رابطه ۱۰ ایجاد می‌شود.

$$C_t' = \tanh\left(W_{hC}h_t + W_{xC}x_t + b_C\right) \qquad (۱۰)$$

ضرب حالت پنهان توسط خروجی در حالت سلول فعلی تولید می‌شود.

$$O_t = h_t \odot \tanh(C_t) \qquad (۱۱)$$

در نتیجه، مدل DNN قادر به دسته‌بندی نمونه آزمایشی جدید به دلیل عدم توانایی تعمیم آن نیست.

۴- نتایج آزمایش

در این بخش، مجموعه داده مورد آزمایش، معیارهای ارزیابی و نتایج آن آورده شده و مقایسه‌ای بین آنها و کارهای قبلی صورت می‌گیرد.

۴-۱- مجموعه داده

بدافزار مبهم بدافزاری است که برای جلوگیری از شناسایی و نابودی پنهان می‌شود. مجموعه داده بدافزار مبهم CIC-MalMem-2022 [۲۲] برای آزمایش روش‌های شناسایی بدافزار از طریق حافظه طراحی شده است. این مجموعه داده برای نشان دادن هر چه بیشتر یک موقعیت واقعی از بدافزاری که در دنیای واقعی رایج است ایجاد شده است. این نرم‌افزار که از بدافزارهای جاسوسی، باج‌افزار و اسب تروجان تشکیل شده است، مجموعه داده متوازنی را تشکیل می‌دهد که می‌تواند برای آزمایش سیستم‌های شناسایی بدافزار مبهم مورد استفاده قرار گیرد.
مجموعه داده بدافزار مبهم بر روی شبیه‌سازی سناریوهای دنیای واقعی تمرکز دارد. جدول ۱ دسته‌بندی داده‌های بدون خطر و دارای خطر و جدول ۲ دسته‌های مختلف موجود در بدافزارهای جاسوسی، باج‌افزار و اسب تروجان را نشان می‌دهد.

جدول ۱- دسته‌بندی کلی داده‌ها در مجموعه داده

نوع داده	تعداد نمونه‌ها
بی‌خطر	۲۹۱۶
جاسوسی	۹۸۲
باج‌افزار	۹۸۶
اسب تروجان	۹۴۸

جدول ۲- دسته‌بندی داده‌های مخرب مجموعه داده با جزئیات

نوع دسته	نوع داده	تعداد نمونه‌ها
جاسوسی	Transponder	۲۴۱
	TIBS	۱۴۱
	180Solutions	۲۰۰
	Coolwebsearch	۲۰۰
	Gator	۲۰۰
باج‌افزار	Ako	۲۰۰
	Shade	۲۲۰
	Conti	۲۰۰
	MAZE	۱۹۵
	Pysa	۱۷۱
اسب تروجان	Scar	۲۰۰
	Reconyc	۱۵۷
	Zeus	۱۹۵
	Emotet	۱۹۶
	Refroso	۲۰۰

۴-۲- معیارهای ارزیابی

ارزیابی رویکرد ارائه شده با پارامترهای دقت، صحت، یادآوری، امتیاز F1 و چند پارامتر دیگر که در رابطه‌های ۱۲ تا ۱۷ نشان داده شده‌اند انجام می‌شود. با تشریح چهار پارامتر مثبت واقعی (TP)، منفی کاذب (FN)، مثبت کاذب (FP) و منفی واقعی (TN) شروع می‌کنیم که برای محاسبه سایر معیارهای ارزیابی به آنها نیاز داریم. TP به تعداد نمونه‌هایی اشاره دارد که به‌درستی به‌عنوان حمله شناسایی شده‌اند. FN تعداد نمونه‌هایی است که داده‌های حمله را به اشتباه به‌عنوان داده عادی طبقه‌بندی می‌شود. FP تعداد موارد عادی را نشان می‌دهد که به‌اشتباه به‌عنوان حمله طبقه‌بندی شده‌اند. TN تعداد نمونه‌هایی را نشان می‌دهد که به‌درستی به‌عنوان عادی طبقه‌بندی شده‌اند. سایر معیارها با استفاده از رابط زیر نشان داده شده‌اند:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (۱۲)$$

$$Precision = \frac{TP}{FP+TP} \quad (۱۳)$$

$$Recall = \frac{TP}{TP+FN} \quad (۱۴)$$

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (۱۵)$$

$$Specificity = \frac{TN}{TN+FP} \quad (۱۶)$$

$$MCC = \frac{TN*TP-FN*FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (۱۷)$$

۴-۳- نتایج به‌دست آمده

این بخش یک ارزیابی دقیق از مدل پیشنهادی ارائه می‌دهد. مدل پیشنهادی بر روی مجموعه داده CIC-MalMem-2022 ارزیابی می‌شود. عملکرد هر دو روش CNN و LSTM برای طبقه‌بندی‌های دودویی و چند کلاسه آزمایش می‌شود. چندین حالت مختلف در تعداد لایه‌ها مورد بررسی قرار گرفته و بر اساس آن، نتایج برای دسته‌های دودویی و چند کلاسه ارائه می‌شود. همین مقایسه با برخی راه حل‌های موجود نیز صورت می‌گیرد.

جدول ۳ مقادیر معیارهای ارزیابی را با در نظر گرفتن بهترین حالت‌ها از تعداد مختلف لایه‌های CNN در مرحله انتخاب ویژگی‌ها در حالت دودویی که با استفاده از زبان برنامه‌نویسی پایتون روی سخت‌افزار با پردازنده 2.5GHZ و رم ۸ گیگابایت اجرا شده نشان می‌دهد. همچنین در همین جدول، مقادیر مربوطه بدون استفاده از CNN نیز نشان داده شده است. همان‌طور که در این جدول مشاهده می‌شود اگر از CNN قبل از مرحله دسته‌بندی داده‌ها استفاده نشود تمامی پارامترهای ارزیابی، مقادیر بسیار کمی خواهند داشت و مدل به هیچ عنوان قابل اعتماد نخواهد بود. همچنین چنانچه برای لایه‌های کانولوشن، جمع‌آوری، مسطح، متراکم و اتصال کامل، تعداد لایه‌ها را به‌ترتیب ۴، ۲، ۳، ۱ و ۳ در نظر بگیریم تمامی مقادیر نسبت به سایر مقادیر، نتایج بهتری را نشان می‌دهند که این موضوع بیانگر آن است که این تعداد لایه‌ها می‌توانند بهترین ویژگی‌ها را از مجموعه داده استخراج کرده و با اجرای روش LSTM بر روی آن به چنین نتایجی دست پیدا کرد.

در ادامه جدول ۴، نتایج را با در نظر گرفتن شرایط ذکر شده اما این بار برای حالت چند کلاسه و برای حملات مختلف نشان می‌دهد. با مشاهده نتایج مشخص می‌شود که بهترین نتایج به‌ترتیب برای تعداد ۴، ۳، ۳، ۲ و ۳ لایه برای لایه‌های کانولوشن، جمع‌آوری، مسطح، متراکم و کاملاً متصل به‌دست می‌آید.

نتایج جدول برای چند کلاسه نسبت به دودویی کمتر است اما باز هم مقادیر بسیار خوبی است.

جدول ۵ نیز مقایسه‌ای از راهکار ارائه شده که از CNN و LSTM برای تشخیص بدافزار استفاده نموده و برخی دیگر از روش‌های یادگیری ماشین را نشان می‌دهد. با توجه به جدید بودن مجموعه داده استفاده شده، به‌ناچار روش‌های یادگیری مختلف را روی آن آزمایش نموده و مقایسه‌ها نیز بر اساس آن صورت گرفته است. به‌عنوان مثال در جدول ۵، روش SVM+RF به معنای آن است که در مرحله کاهش ویژگی‌ها از SVM (ماشین بردار پشتیبان) و در مرحله دسته‌بندی از جنگل تصادفی (Random Forest) استفاده شده است. مقایسه‌ها نشان می‌دهد که روش پیشنهادی نسبت به سایر کارها از برتری قابل توجهی برخوردار است.

جدول ۳- نتایج آزمایش دودودیی با تعداد مختلف لایه‌های CNN

لایه کانولوشن	لایه جمع-آوری	لایه مسطح	لایه متراکم	لایه متصل	Accuracy	Precision	Recall	F1 Score	MCC
۱	۱	۱	۱	۱	۹۴.۲۵	۹۸.۸۳	۹۸.۰۱	۹۶.۲۷	۸۸.۵۵
۲	۱	۲	۱	۲	۹۸.۱۸	۹۷.۷۳	۹۷.۲۲	۹۴.۵۴	۸۲.۵۲
۳	۳	۲	۲	۲	۹۹.۷۹	۹۸.۶۷	۹۶.۲۸	۹۸.۴۳	۸۵.۳۹
۴	**۲**	**۳**	**۲**	**۳**	**۹۹.۹۹**	**۹۹.۷۵**	**۹۹.۵۸**	**۹۸.۴۸**	**۹۰.۲۶**
صفر	صفر	صفر	صفر	صفر	۷۸.۰۷	۶۵.۶۴	۷۴.۵۱	۶۶.۴۸	۶۱.۱۷

جدول ۴- نتایج میانگین چند کلاسه با تعداد لایه‌های مختلف CNN

لایه کانولوشن	لایه جمع-آوری	لایه مسطح	لایه متراکم	لایه متصل	Accuracy	Precision	Recall	F1 Score	MCC
۱	۱	۳	۳	۳	۹۰.۷۳	۹۳.۲۴	۸۸.۲۷	۸۶.۴۴	۸۷.۲۹
۲	۳	۲	۳	۲	۸۸.۲۷	۸۹.۷۳۳	۹۰.۵۱	۸۹.۵۴۲	۷۲.۴۳
۳	۳	۳	۲	۳	۸۲.۳۶	۸۸.۷۱	۸۵.۲۲	۹۰.۸۴	۸۱.۳۷
۴	**۳**	**۳**	**۲**	**۳**	**۹۹.۷۱**	**۹۵.۵۲**	**۹۶.۴۶**	**۹۴.۴۴**	**۸۸.۵۲**
صفر	صفر	صفر	صفر	صفر	۶۸.۲۷	۶۳.۷۳	۷۱.۵۷	۵۹.۴۹	۶۰.۱۴

جدول ۵- مقایسه روش پیشنهادی با برخی دیگر از راهکارها

مرحله کاهش ویژگی	مرحله طبقه‌بندی	Accuracy	Precision	Recall	F1 Score	MCC
ماشین بردار پشتیبان [۱۰]	جنگل تصادفی	۹۷.۵۴	۹۷.۸۳	۹۸.۴۹	۹۸.۷۵	۸۸.۱۰
شبکه عصبی کانولوشن [۱۲]	رمزگذار خودکار عمیق DAE	۹۸.۶۴	۹۸.۴۵	۹۵.۰۹	۹۷.۱۴	۸۶.۵۴
استخراج ویژگی [۱۳]	یادگیری عمیق	۹۸.۳۳	۹۵.۸۶	۹۷.۵۲	۹۴.۷۱	۸۷.۵۵
یادگیری عمیق[۱۴]	رمزگذار Stacked Denoising	۹۹.۹۶	۹۸.۳۴	۹۹.۲۷	۹۸.۳۳	۸۷.۲۱
شبکه عصبی عمیق [۱۶]	Android MobiTive	۹۸.۷۶	۹۷.۳۶	۹۷.۴۲	۹۸.۲۸	۸۷.۷۴
شبکه عصبی کانولوشن	شبکه حافظه طولانی کوتاه‌مدت	۹۹.۹۹	۹۹.۷۵	۹۹.۵۸	۹۸.۴۸	۹۰.۲۶

۵- نتیجه‌گیری

با تکامل شبکه و اینترنت، تولیدکنندگان بدافزار به‌سرعت کدهای مخرب خود را تطبیق داده و اغلب آنها برای سوء استفاده از آسیب‌پذیری‌ها در سیستم‌عامل استفاده می‌شوند. اگرچه روش‌های زیادی برای شناسایی بدافزار مبهم یا پنهان بر اساس تجزیه و تحلیل حافظه وجود دارد، اما دقت کافی برای تشخیص آنها در هر دو حالت دودویی و چند کلاسه مهیا نشده است. به‌عنوان راه حلی برای این مشکل، مدلی از شناسایی بدافزار در این مقاله پیشنهاد شده است که از شبکه عصبی کانولوشن برای استخراج ویژگی های مهم مجموعه داده استفاده می‌کند تا در مرحله طبقه‌بندی، کمترین میزان خطا به‌دست آید. در مرحله طبقه‌بندی نیز از روش LSTM که یک روش یادگیری عمیق است استفاده می‌شود تا بالاترین دقت تشخیص بدافزار برای هر دو حالت دودویی (۹۹..٪۹۹) و چندکلاسه (۷۱..٪۹۹) به‌دست آید. برای این کار از یک مجموعه داده بسیار جدید به نام MalMemAnalysis-2022 که بدافزارهای آن در سه دسته اصلی، بدافزارهای جاسوسی، باج‌افزار و بدافزار اسب تروجان قرار می‌گیرند استفاده شده است که این یک برتری نسبت به سایر روش‌های ارائه شده است.

با استفاده از مجموعه داده ذکر شده و انجام آزمایش مشخص شد که مدل نهایی بر اساس معیارهای ارزیابی مختلف روشی بسیار کارآمد و قابل اعتماد است. این نتیجه زمانی به‌دست آمده است که روش پیشنهادی را با روش هایی از قبیل درخت تصمیم، بردار پشتیبان، رگرسیون و شبکه عصبی و غیره به‌عنوان ابزارهای تشخیص و کاهش ویژگی‌ها مقایسه کردیم. نتایج مقایسه نشان داد که مدل پیشنهادی، دقت طبقه‌بندی بیشتر و عملکرد بهتری دارد.

تقدیر و تشکر:

نویسندگان برای تهیه این مقاله از امکانات نرم‌افزاری و سخت‌افزاری دانشکده‌های علوم دانشگاه گلستان و فنی مهندسی چمران گرگان استفاده نموده‌اند. لذا نویسندگان، مراتب تقدیر و تشکر خود را از مسئولان محترم این دو دانشکده که همکاری‌های لازم را به‌عمل آورده‌اند اعلام می‌دارند.

تعارض منافع:

نویسندگان اعلام می‌کنند که در مورد انتشار این مقاله تعارض منافع وجود ندارد.

تاییدیه اخلاقی:

نویسندگان متعهد می‌شوند که مطالب این مقاله را در هیچ مجله دیگری به چاپ نرسانده‌اند.

مشارکت‌های نویسندگان:

علی‌اکبر تجری سیاه‌مرزکوه: روش‌شناسی، موارد نرم‌افزار، اعتبارسنجی، تحقیق، نگارش پیش‌نویس مطالب، بررسی و ویرایش

علی رحیمی حسین‌آباد: روش‌شناسی، تحقیق، نگارش متن، بررسی و ویرایش محتوا، منابع

منابع مالی:

در انجام پژوهش هیچ منابع مالی مورد استفاده قرار نگرفته است.

مراجع

[1] A.S. Shatnawi, Y. Qussai, and Y. Abdulrahman. “An Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms.” *The 3rd International Workshop on DataDriven Security DDSW* (2022): 22-25.

[2] N.S. Escanilla, L. Hellerstein, R. Kleiman, Z. Kuang, J. Shull, and D. Page. “Recursive feature elimination by sensitivity testing.” *17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE*, (2018): 40–47.

[3] L. Taheri. A.F.A. Kadir, and A. Habibi. “Extensible android malware detection and family classification using network-flows and api-calls.” *International Carnahan Conference on Security Technology (ICCST)*, IEEE, (2019): 1–8.

[4] S. Smmarwar, G. Gupta, S. Kumar, and P. Kumar. “An optimized and efficient android malware detection framework for future sustainable computing.” *Sustainable Energy Technologies and Assessments* 1 (2022): 1-8.

[5] S.I. Imtiaz, S.A.R. Javed, Z. Jalil, X. Liu, and W. Alnumay. “DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network.” *Future Generation Computer Systems* 1 (2021): 844-856.

[6] E. Parsaeimehr, M. Fartash, and J.A. Torkestani. “An ensemble deep learning model to enhance feature representation for entity detection.” *Journal of Modeling in Engineering* 20 (2022): 103-112. (in Persian)

[7] E. Berenjkar. “Evaluation of the performance of artificial neural networks integrated with whale optimization and ant colony optimization algorithms in estimating the drilling rate of penetration and compare with simple neural networks and mathematical conventional models.” *Journal of Modeling in Engineering* 19 (2021): 115-135. (in Persian)

[8] P. Bhat, and K. Dutta. “A multi-tiered feature selection model for android malware detection based on Feature discrimination and Information Gain.” *Journal of King Saud University-Computer and Information Sciences* 34 (2022): 9464-9477.

[9] S. Mahdavifar, A. Kadi, R. Fatemi, D. Alhadidi, and A.A. Ghorbani. “Dynamic android malware category classification using semi-supervised deeplearning.” *IEEE Intl Conf on Dependable, Autonomic and Secure Computing* (2020): 515–522.

[10] C. Li, K. Mills, D. Niu. R. Zhu, H. Zhang, and H. Kinawi. “Android malware detection based on factorization machine.” IEEE Access 7 (2019): 184008–184019.

[11] C. Zhao, W. Zheng, L. Gong, M. Zhang, and C. Wang. “Quick and accurate android malware detection based on sensitive APIs.” *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, IEEE, (2018): 143–148.

[12] W. Wang, M. Zhao, and J. Wang. “Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network.” *Journal of Ambient Intelligence and Humanized Computing* 10 (2018): 3035–3043.

[13] T. Kim, B. Kang, M. Rho, S. Sezer, and E.G. Im. “A multimodal deep learning method for android malware detection using various features.” *IEEE Transactions on Information Forensics and Security* 14 (2018): 773–788.

[14] J. Xu, Y. Li, and R. Deng. “Differential Training: A Generic Framework to Reduce Label Noises for Android Malware Detection.” *2021 Network and Distributed System Security Symposium* (2021): 1-14.

[15] C. Hasegawa, and H. Iyatomi. “One-dimensional convolutional neural networks for Android malware detection.” 14th International Colloquium on Signal Processing & its Applications (CSPA). IEEE. (2018): 99–102.

[16] R. Feng, S. Chen, X. Xie, G. Meng, S.W. Lin, and Y. Liu. “A performance-sensitive malware detection system using deep learning on mobile devices.” IEEE Transactions on Information Forensics and Security 16 (2020): 1563–1578.

[17] J. Zhang, Q. Jixin, Y. Zheng, H. Yin, L. Ou, and K. Zhang. “A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding.” *Computer Security* 84 (2019): 376–392.

[18] Y. Zhang, Y. Yang, and X. Wang. “A novel android malware detection approach based on convolutional neural network.” *2nd International Conference on Cryptography and Security Privacy*, New York, NY, USA, ACM. (2018): 144–149.

[19] S.A. Khowaja, and P. Khuwaja. “Q-learning and LSTM based deep active learning strategy for malware defense in industrial IoT applications.” Multimedia Tools and Applications 80 (2021): 14637–14663.

[20] K. Xu, Y. Li. R.H. Deng, and K. Chen. “DeepRefiner: multi-layer android malware detection system applying deep neural networks.” IEEE European Symposium on Security and Privacy, IEEE. (2018): 473–487.

[21] S.J. Hussain, U. Ahmed, H. Liaquat, S. Mir, NZ. Jhanjhi, and M. Humayun. “IMIAD: intelligent malware identification for android platform.” International Conference on Computer Information Science, IEEE. (2019): 1–6.

[22] T. Carrier, P. Victor, A. Tekeoglu, and A.H. Lashkari. “Detecting Obfuscated Malware using Memory Feature Engineering.” The 8th International Conference on Information Systems Security and Privacy (ICISSP) (2022): 177-188.