



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 Hossein Abad ^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>

© 2025 Published by Semnan University Press.

This is an open access article under the CC-BY 4.0 license.(<https://creativecommons.org/licenses/by/4.0/>)

* 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

تشخیص بدافزار در سیستم عامل اندروید با استفاده از شبکه عصبی کانولوشن و شبکه حافظه طولانی کوتاه مدت

علی اکبر تجری سیاهمرزکوه^{۱*}، علی رحیمی حسین آباد^۲

اطلاعات مقاله	چکیده
دریافت مقاله: ۱۴۰۲/۰۷/۰۷ بازنگری مقاله: ۱۴۰۳/۰۳/۰۹ پذیرش مقاله: ۱۴۰۳/۰۳/۲۲	استفاده از تلفن های همراه با سیستم عامل اندروید روز به روز در حال گسترش است. سیستم عامل اندروید به خودی خود ابزار قدرتمندی برای تشخیص بدافزار ندارد. از این رو، مهاجمان به راحتی از طریق گوشی تلفن همراه افراد وارد حریم خصوصی آنها شده و آنها را در معرض خطر جدی قرار می دهند. تاکنون تحقیقات زیادی بر روی تشخیص بدافزار صورت گرفته است. یکی از مشکلات عمده این راهکارها، دقت پایین در تشخیص چند کلاسه روی مجموعه داده ها 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 لایه های مختلف، سعی کرده ایم تا حداکثر تعداد ویژگی های مهم را از مجموعه داده استخراج نماییم. در فاز طبقه بندی داده ها نیز از الگوریتم یادگیری شبکه حافظه طولانی کوتاه مدت (LSTM) استفاده می کنیم تا با آزمایش آن بر روی ویژگی های انتخاب شده، داده ها با حداکثر دقت ممکن طبقه بندی شوند. نتایج آزمایش بر روی مجموعه داده جدید MalMemAnalysis-2022 نشان می دهد که استفاده از این دو الگوریتم و تغییر در تعداد لایه ها می تواند در بهترین حالت به ترتیب منجر به دقت های ۹۹٪.۷۱ و ۹۹٪.۹۹ در دسته بندی دودویی و چند کلاسه در تشخیص بدافزار شود که نسبت به روش های موجود برتری دارد.

واژگان کلیدی:

تشخیص بدافزار،

شبکه عصبی کانولوشن

(CNN)،

شبکه حافظه طولانی کوتاه -

مدت (LSTM)،

مجموعه داده

MalMemAnalysis-2022

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

© 2025 Published by Semnan University Press.

This is an open access article under the CC-BY 4.0 license. (<https://creativecommons.org/licenses/by/4.0/>)

۱- مقدمه

استفاده فراگیر از تلفن های همراه، آنها را به بخشی ضروری از فعالیت های روزمره زندگی تبدیل کرده است [۱]. اندروید بدون شک نسبت به سایر سیستم عامل های موجود محبوبیت بیشتری در بازار به دست آورده است. از طرفی، استفاده فراگیر از اندروید توجه بیشتر مهاجمان سایبری را نیز به خود جلب کرده است. نقص های امنیتی در گوشی های

هوشمند اندرویدی نشان می دهد که آنها در برابر حملات فیشینگ پیامکی بسیار آسیب پذیر هستند. معماری باز سیستم عامل اندروید نیز، اثرات مخرب جرایم سایبری بر پلتفرم اندروید را تشدید می کند [۲]. نگرانی های اصلی امنیت سایبری شامل حملات بدافزار، عدم سرویس دهی، بات نت ها، روت کیت ها، نفوذها و باج افزارها است. بدافزار نرم افزاری است که شامل ویروس ها، 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 حرفه ای، 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 کوتاه مدت. مدل سازی در مهندسی، ۲۳(۸۰)، ۴۷-۵۷. doi: 10.22075/jme.2024.31932.2537

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 آنجایی که تعداد ویژگی‌های موجود در فایل‌های اندرویدی بسیار زیاد است و سربار محاسباتی و از طرفی کاهش دقت سیستم تشخیص نفوذ را به دنبال 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

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 به دلیل 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 با 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 آنجایی که فایل‌های apk به راحتی در دسترس کاربران اندروید قرار می‌گیرند، به منبعی برای حملات تبدیل شده‌اند. از طرفی، اخلاص‌گران از مجوزهایی که توسط خدمات گوگل پلی 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 به صورت 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 به خطر می‌اندازد و اگر هم کاربران، این مجوزها را رد کنند نمی‌توانند به درستی 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 به صورت 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

تجزیه و تحلیل مجوزها را ارائه نمودند. با بررسی هایی که در کار تحقیقاتی آنها صورت گرفت، طبقه بندی کننده 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 برنامه های متعدد به عنوان بردار ویژه در طول فرآیند آموزش با استفاده از تحلیل معکوس بازیابی می شوند. نتایج آزمایش نشان می دهد که مقادیر پارامتر دقت و صحت این روش به ۹۳ درصد می رسد.

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 نرم افزارهای مخرب، چندین CNN برای جستجوی برنامه های مخرب اندروید و بازسازی ویژگی ها استفاده شده است. این روش در مقایسه با روش های یادگیری ماشین سنتی، بهبود قابل توجهی در شناسایی نرم افزارهای مخرب در اندروید نشان می دهد. به طور مشخص، کارایی این مدل ۵ درصد بهتر از ماشین بردار پشتیبان است. کیم و همکاران [۱۳]، یک معماری منحصر به فرد برای شناسایی نرم افزارهای مخرب در اندروید پیشنهاد کردند. به منظور اصلاح ویژگی ها و به دست آوردن ویژگی های مؤثرتر برای تشخیص بدافزار، یک رویکرد استخراج ویژگی مبتنی بر شباهت ارائه شده است. چارچوب پیشنهادی از ویژگی های مختلف برای به تصویر کشیدن ویژگی های برنامه های اندروید از دیدگاه های مختلف استفاده می کند. چندین مرحله آموزش بر روی نمونه داده ها به منظور ارزیابی عملکرد فاز آموزش صورت گرفته است.

ژو و همکاران [۱۴]، روش یادگیری ویژگی بدون نظارت را معرفی کردند که برای آموزش ویژگی های مؤثر به عنوان اولین گام در چارچوب تشخیص وپروس استفاده می شود. 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. به منظور بهبود قابلیت های تشخیص بدافزار، ویژگی های مهم تر و مؤثرتر استخراج شده و برای آموزش مدل تشخیص بدافزار از روش های طبقه بندی مانند ماشین بردار پشتیبان یا k- نزدیکترین همسایگی استفاده شده است.

تشخیص نفوذ تأثیر مثبت بگذارد. هدف این مقاله استخراج مجموعه ای از ویژگی های مؤثر یا ویژگی های مرتبط است. مجموعه ویژگی های انتخاب شده می تواند تجزیه و تحلیل دقیق تری را ارائه دهد و منجر به ارائه یک مدل تشخیص بسیار دقیق و قابل اعتماد شود. در این مقاله، تلاش می کنیم تا مجموعه ویژگی های مرتبط را با استفاده از کاهش ویژگی ها ایجاد نموده و آن را برای عملیات طبقه بندی آماده کنیم.

در این مقاله، یک مدل انتخاب ویژگی پیشنهاد می شود که می تواند ویژگی های مرتبط را از یک مجموعه وسیع استخراج نماید. این مدل از دو بخش تشکیل شده است که بخش اول کاهش ویژگی ها و بخش دوم آزمایش بر روی ویژگی های انتخاب شده است.

در روش پیشنهادی، ابتدا ویژگی ها به عنوان ورودی دریافت شده و سپس به یک فاز کاهش ویژگی ارسال می شوند. در گام اول، ویژگی هایی که از لحاظ رفتاری با یکدیگر وابستگی دارند شناسایی شده و حذف می شوند. همچنین برخی از ویژگی ها در طبقه بندی مفید نیستند، زیرا وابستگی کمی با مقادیر ویژگی کلاس دارند، از این رو این دسته نیز حذف می شوند.

در گام بعدی، با استفاده از شبکه حافظه طولانی کوتاه مدت (LSTM) که یک معماری شبکه عصبی بازگشتی است اقدام به طبقه بندی داده ها می کنیم. این ابزار قدرتمند مبتنی بر یادگیری عمیق، بر خلاف شبکه عصبی سنتی دارای اتصالات بازخوردی است که به آن اجازه می دهد تا کل توالی داده ها 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، کارایی روش پیشنهادی بررسی می شود [۹].

ادامه مقاله به شرح زیر است: بخش ۲ کارهای مرتبط را مورد بحث قرار می دهد. بخش ۳ رویکرد پیشنهادی را نشان می دهد و بخش ۴ به بررسی معیارهای ارزیابی و نتایج و تجزیه و تحلیل می پردازد. بخش ۵ تجزیه و تحلیل با رویکرد مقایسه ای از نتایج را نشان می دهد.

۲- پیشینه تحقیق

در این بخش، برخی از کارهای پیشین انجام شده در حوزه تشخیص حملات اندرویدی را بررسی می کنیم. لی و همکاران [۱۰]، یک ابزار تشخیص بدافزار مبتنی بر

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.

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

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 صورتی که 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 زمانی که 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.

$$v' = \frac{v - \mu_A}{\sigma_A} \quad (2)$$

۳-۲- انتخاب ویژگی‌ها

با هدف به حداقل رساندن پیچیدگی مدل تشخیص حملات

هاسگاو 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 $97\% \pm 0.04$ 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 آماده‌سازی 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 کارایی.



شکل ۱- فلوچارت کلی روش پیشنهادی

شکل (۱)، فلوچارت کلی روش پیشنهادی را نشان می‌دهد.

رابطه ۳ و ۴، عملکرد لایه کانولوشن را نشان می دهد که در آن x_k ورودی لایه است. خروجی لایه قبل با s_i نشان داده می شود و w_{ik} نشان دهنده هسته از i تا k است. b_k مقدار بایاس نرون در لایه کانولوشن است. تابع فعال سازی ReLU با $f(O)$ نشان داده می شود. رابطه ۵، ReLU را توصیف می کند. y_k خروجی لایه کانولوشن بعدی است.

$$f(x_k) = \max(0, x_k) \quad (5)$$

۳-۳- طبقه بندی داده ها

LSTM (حافظه طولانی کوتاه مدت) یک معماری شبکه عصبی بازگشتی (RNN) است که به طور گسترده در یادگیری عمیق استفاده می شود. این روش برای یافتن وابستگی های بلندمدت، عالی و برای کارهای پیش بینی توالی ایده آل است. برخلاف شبکه های عصبی سنتی، LSTM دارای اتصالات بازخوردی است که به آن اجازه می دهد تا کل توالی داده ها و نه فقط نقاط داده ای جداگانه را پردازش کند، این امر آن را در درک و پیش بینی الگوها در داده های متوالی مانند سری های زمانی، متن و گفتار بسیار مؤثر می کند.

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

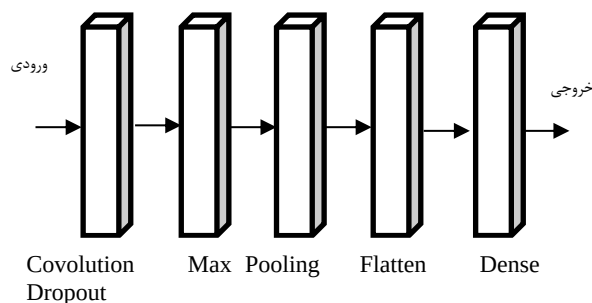
بخش اول انتخاب می کند که آیا اطلاعاتی که از مهر زمانی قبلی به دست می آید باید به خاطر سپرده شود یا نامربوط است و می تواند فراموش شود. تابع فعال سازی از طریق رابطه ۶ محاسبه می شود به گونه ای که در آن b بردار بایاس، h_{t-1} خروجی بلوک قبلی، x_t دنباله ورودی و W_{xf} و W_{hf} به ترتیب، ماتریس وزن بردارهای ورودی و خروجی از سلول های قبلی و فعلی را نشان می دهند. σ نیز توسط تابع سیگموئید γ ارائه می شود.

$$f_t = \sigma(W_{hf}h_{t-1} + W_{xf}x_t + b_f) \quad (6)$$

$$\sigma(x) = (1 + e^{-x})^{-1} \quad (7)$$

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

بدافزار و حفظ دقت تشخیص بالا، کاهش ویژگی در مدل تشخیص حمله پیشنهادی را بر اساس معماری CNN طراحی کرده ایم که شامل لایه های کانولوشن، حداکثر جمع آوری (max-pooling) مسطح، متراکم و لایه های کاملاً متصل است. شکل (۲)، شبکه کانولوشن پیشنهادی برای انتخاب ویژگی های مؤثر را نشان می دهد.



شکل ۲- نمایش شبکه عصبی کانولوشن برای استخراج ویژگی ها

CNN یک روش یادگیری عمیق است که از لایه های مختلفی مانند کانولوشن، لایه های ادغام و لایه های کاملاً متصل تشکیل شده است. CNN معمولاً برای طبقه بندی تصویر و تشخیص صدا استفاده می شود. در این مقاله از CNN برای آماده سازی داده ها جهت شناسایی فعالیت های مخرب در اندروید استفاده می کنیم.

در بخش نتایج آزمایش با تغییر تعداد هر سطح از لایه ها اقدام به بررسی میزان دقت مدل می کنیم. به طور مثال، شبکه ای 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) می باشد که مقدار اول، تعداد پویای نمونه ها و مقدار دوم، تعداد ویژگی های ورودی است. ۴۰ فیلتر در دو لایه اول کانولوشن استفاده شده است که خروجی به شکل (۱۶ و unknown) تولید می کند. خروجی لایه های کانولوشن به عنوان ورودی به لایه max-pooling وارد می شود. 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 لایه ها نه تنها مهم ترین ویژگی ها را استخراج می کند بلکه نویزهای احتمالی را نیز کاهش می دهد.

$$x_k = b_k + \sum_{i=1}^N(s_i, w_{ik}) \quad (3)$$

$$y_k = f(x_k) \quad (4)$$

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

$$O_t = h_t \odot \tanh(C_t) \quad (11)$$

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 متوازی 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.

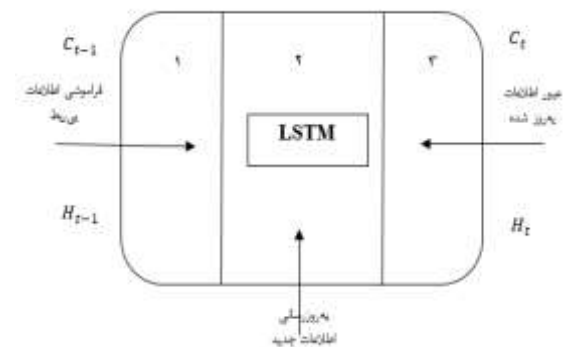
جدول ۱- دسته‌بندی کلی داده‌ها در مجموعه داده

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

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:

$$O_t = \sigma(W_{ho}h_{t-1} + W_{xo}x_t + b_o) \quad (8)$$

وضعیت فعلی سلول با یکپارچه‌سازی به‌همراه ورودی محاسبه می‌شود:

$$C_t = f_t \odot C_{t-1} + i_t \odot C_t' \quad (9)$$

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.

$$C_t' = \tanh(W_{hc}h_t + W_{xc}x_t + b_c) \quad (10)$$

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

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

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

۳-۴- نتایج به دست آمده

این بخش یک ارزیابی دقیق از مدل پیشنهادی ارائه می دهد. مدل پیشنهادی بر روی مجموعه داده CIC-MalMem-2022 ارزیابی می شود. عملکرد هر دو روش CNN و LSTM برای طبقه بندی های دودویی و چند کلاسه آزمایش می شود. چندین حالت مختلف در تعداد لایه ها مورد بررسی قرار گرفته و بر اساس آن، نتایج برای دسته های دودویی و چند کلاسه ارائه می شود. همین مقایسه با برخی راه حل های موجود نیز صورت می گیرد. جدول ۳ مقادیر معیارهای ارزیابی را با در نظر گرفتن بهترین حالت ها از تعداد مختلف لایه های CNN در مرحله انتخاب ویژگی ها در حالت دودویی که با استفاده از زبان برنامه نویسی پایتون روی سخت افزار با پردازنده 2.5GHZ و رم ۸ گیگابایت اجرا شده نشان می دهد. همچنین در همین جدول، مقادیر مربوطه بدون استفاده از CNN نیز نشان داده شده است. همان طور که در این جدول مشاهده می شود اگر از CNN قبل از مرحله دسته بندی داده ها استفاده نشود تمامی پارامترهای ارزیابی، مقادیر بسیار کمی خواهند داشت و مدل به هیچ عنوان قابل اعتماد نخواهد بود. همچنین چنانچه برای لایه های کانولوشن، جمع آوری، مسطح، متراکم و اتصال کامل، تعداد لایه ها را به ترتیب ۴، ۲، ۳، ۱ و ۳ در نظر بگیریم تمامی مقادیر نسبت به سایر مقادیر، نتایج بهتری را نشان می دهند که این موضوع بیانگر آن است که این تعداد لایه ها می توانند بهترین ویژگی ها 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 می دهد. با مشاهده نتایج مشخص می شود که بهترین نتایج به ترتیب برای تعداد ۴، ۳، ۳، ۲ و ۳ لایه برای لایه های کانولوشن، جمع آوری، مسطح، متراکم و کاملاً متصل به دست می آید.

جدول ۲- دسته بندی داده های مخرب مجموعه داده با جزئیات

نوع دسته	نوع داده	تعداد نمونه ها
جاسوسی	Transponder	۲۴۱
	TIBS	۱۴۱
	180Solutions	۲۰۰
	Coolwebsearch	۲۰۰
	Gator	۲۰۰
باج افزار	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 رابطه های ۱۲ تا ۱۷ نشان داده شده اند انجام می شود. با تشریح چهار پارامتر مثبت واقعی (TP)، منفی کاذب (FN)، مثبت کاذب (FP) و منفی واقعی (TN) شروع می کنیم که برای محاسبه سایر معیارهای ارزیابی به آنها نیاز داریم. TP به تعداد نمونه هایی اشاره دارد که به درستی به عنوان حمله شناسایی شده اند. FN تعداد نمونه هایی است که داده های حمله را به اشتباه به عنوان داده عادی طبقه بندی می شود. FP تعداد موارد عادی را نشان می دهد که به اشتباه به عنوان حمله طبقه بندی شده اند. TN تعداد نمونه هایی را نشان می دهد که به درستی به عنوان عادی طبقه بندی شده اند. سایر معیارها با استفاده از رابط زیر نشان داده شده اند:

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

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

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

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 دودویی با تعداد مختلف لایه‌های CNN

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

۵- نتیجه گیری

با تکامل شبکه و اینترنت، تولیدکنندگان بدافزار به سرعت کدهای مخرب خود را تطبیق داده و اغلب آنها برای سوء استفاده از آسیب پذیری ها در سیستم عامل استفاده می شوند. اگرچه روش های 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. به عنوان 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 می کند تا در مرحله طبقه بندی، کمترین میزان خطا به دست آید. در مرحله طبقه بندی نیز از روش LSTM که یک روش یادگیری عمیق است استفاده می شود تا بالاترین دقت تشخیص بدافزار برای هر دو حالت دودویی (۹۹٪/۹۹) و چندکلاسه (۹۹٪/۷۱) به دست آید. برای این کار از یک مجموعه داده بسیار جدید به نام MalMemAnalysis-2022 که بدافزارهای آن در سه دسته اصلی، بدافزارهای جاسوسی، باج افزار و بدافزار اسب تروجان قرار می گیرند استفاده شده است که این یک برتری نسبت به سایر روش های 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 به دست آمده است که روش پیشنهادی را با روش هایی از قبیل درخت تصمیم، بردار پشتیبان،

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.

رگرسیون و شبکه عصبی و غیره به عنوان ابزارهای تشخیص و کاهش ویژگی ها مقایسه کردیم. نتایج مقایسه نشان داد که مدل پیشنهادی، دقت طبقه بندی بیشتر و عملکرد بهتری دارد.

تقدیر و تشکر:

نویسندگان برای تهیه این مقاله از امکانات نرم افزاری و سخت افزاری دانشکده های علوم دانشگاه گلستان و فنی مهندسی چمران گرگان استفاده نموده اند. لذا نویسندگان، مراتب تقدیر و تشکر خود را از مسئولان محترم این دو دانشکده که همکاری های لازم را به عمل آورده اند اعلام می دارند.

تعارض منافع:

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

تاییدیه اخلاقی:

نویسندگان متعهد می شوند که مطالب این مقاله را در هیچ مجله دیگری به چاپ نرسانده اند.

مشارکت های نویسندگان:

علی اکبر تجری سیاه مرزکوه: روش شناسی، موارد نرم افزار، اعتبارسنجی، تحقیق، نگارش پیش نویس مطالب، بررسی و ویرایش

علی رحیمی حسین آباد: روش شناسی، تحقیق، نگارش متن، بررسی و ویرایش محتوا، منابع

منابع مالی:

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

- [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. Berenjkari. "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, N.Z. 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.