Electronical & Cyber Defence
Vol. 13, No. 1, Spring 2025, Serial No. 49
9

Malware detection using federated learning and incremental learning

M.A. Eftekhari, *, M.. Yousef Sanati, M.Mansoorizadeh
*Assistant Professor, Bu-Ali Sina University, Hamedan, Iran

Received:2024 /12/02, Revised: 2025/01/24, Accepted: 2025/03/03, Published: 2025/04/21

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1404.13.1.9.4

ABSTRACT

Android-based mobile devices are widely used due to their ease of use among users. Individuals perform various tasks on their mobile phones, such as banking activities, social networking, and diverse business systems, thereby exposing considerable personal information to risks due to the vulnerabilities of the Android operating system. The rapid development of Android malware has rendered many traditional malware detection methods less accurate over time. Research indicates that machine learning is an effective approach for detecting malware. The rapid evolution of malware contributes to the degradation of accuracy in trained models over time. Moreover, the collection of malware-related data from Android devices jeopardizes users' privacy. To address these issue, this paper employs federated and incremental learning. Recently, federated learning has been introduced for training machine learning models on decentralized devices with the aim of preserving privacy. This study utilizes a Multi-Layer Perceptron (MLP) within the framework of federated learning. Stacking, a type of ensemble learning, is employed for incremental learning. The CICMalDroid 2020 dataset is utilized in this research, using static data to develop the final model. The outcome of this study is a model with an accuracy of 96.49%, demonstrating significant improvement in computational time complexity along with maintaining the quality of learning and model accuracy compared to existing methods.

Keywords: malware detection, machine learning, federated learning, incremental learning, distribution

❖ Corresponding Author Email: mysanati@basu.ac.ir

"پدافند الکترونیکی و سایبری"

سال سیزدهم، شماره ۱، بهار ۱۴۰۴، ص۱۳۰-۱۱۷

شاپا الکترونیکی: ۸۹۷۹-۲۹۸۰ شاپا چاپی: ۴۳۴۷-۲۳۲۲

علمی – پژوهشی

تشخیص بدافزار با استفاده از یادگیری ائتلافی و افزایشی

محمدعلی افتخاری [۱]، مرتضی یوسف صنعتی [۲*]، محرم منصوری زاده [۳]

۱- دانشجوی کارشناسی ارشد، ۲- استادیار، ۳- دانشیار، دانشگاه بوعلی سینا،همدان، ایران

(دریافت: ۱۴۰۳/۰۹/۱۸، بازنگری: ۱۴۰۳/۱۲/۰۱، پذیرش: ۱۴۰۳/۱۲/۲۳، انتشار: ۱۴۰۳/۰۲/۰۱)

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1404.13.1.9.4

چکیده

دستگاه‌های تلفن همراه مبتنی بر اندروید به دلیل راحتی در استفاده کاربران بسیار زیادی دارند. افراد در تلفن‌های همراه خودکارهای مختلفی از جمله فعالیت‌های بانکی، فعالیت در شبکه‌های اجتماعی و سامانه‌های متعدد و متنوع کسب‌وکار را انجام می‌دهند و به همین دلیل اطلاعات شخصی زیادی از آن‌ها به دلیل آسیب‌پذیری سیستم‌عامل اندروید در معرض خطر قرار می‌گیرد. به دلیل توسعه سریع بدافزارهای اندرویدی، بسیاری از روش‌های سنتی تشخیص بدافزار دقت خود را ازدست‌داده‌اند. تحقیقات نشان می‌دهند یادگیری ماشین یک روش مؤثر برای تشخیص بدافزارها است. توسعه سریع بدافزارها باعث می‌شود دقت مدل‌های یادگیری شده بعد از مدتی کاهش پیدا کند. همچنین با جمع-آوری داده‌های مربوط به بدافزارها از دستگاه‌های اندرویدی حریم خصوصی کاربران به خطر می‌افتد. برای حل این مشکل در این مقاله از یادگیری افزایشی و ائتلافی (فدرال) استفاده شده است. اخیراً یادگیری ائتلافی برای آموزش مدل‌های یادگیری ماشین در دستگاه‌های غیرمتمرکز باهدف حفظ حریم خصوصی معرفی شده است. این مقاله از شبکه عصبی (MLP) در چارچوب یادگیری ائتلافی استفاده نموده است. برای یادگیری افزایشی از روش پشته‌ای که یکی از انواع یادگیری جمعی است استفاده شده است. در این پژوهش از مجموعه‌داده CICMalDroid 2020 استفاده شده و با استفاده از داده‌های ایستا، مدل نهایی ایجاد شده است. حاصل این پژوهش مدلی بادقت ۹۶/۴۹ است که مقایسه آن با روش‌های موجود نشانگر بهبود قابل‌توجه پیچیدگی زمانی محاسبات به همراه حفظ کیفیت یادگیری و دقت مدل‌هاست.

کلیدواژه‌ها: تشخیص بدافزار، یادگیری ماشین، یادگیری ائتلافی، یادگیری افزایشی، توزیع‌شدگی

* رایانامه نویسنده مسئول: mysanati@basu.ac.ir

۱. مقدمه

اندروید به دلیل انعطاف‌پذیری، استحکام و پشتیبانی سخت‌افزاری خود در ارتباط با انواع حسگرها در حوزه اینترنت اشیا (IoT) بسیار موردتوجه قرار گرفته است. ازاین‌رو بدافزارهای اندرویدی در IoT می‌توانند آسیب‌های جدی به کاربران وارد کنند [۱]. از جمله این آسیب‌ها می‌توان به موارد زیر اشاره کرد: نقض داده‌های شخصی کاربران[۲, ۳]، کنترل دستگاه‌های IoT توسط مهاجمان [۴, ۵] و ایجاد اختلال در عملکرد دستگاه‌هایIoT [۶]. همچنین بیش‌تر دستگاه‌های IoT از آنتی‌ویروس یا اپلیکیشن‌های تشخیص بدافزار استفاده نمی‌کنند؛ بنابراین، تشخیص سریع و بادقت بدافزار اندروید در IoT امری ضروری است تا از وقوع این آسیب‌ها جلوگیری شود. علاوه بر این در سال‌های اخیر، اکثر دستگاه‌های اینترنت اشیا صنعتی(IIoT) از سیستم‌عامل اندروید به‌عنوان محبوب‌ترین و شناخته‌شده‌ترین سیستم‌عامل تلفن همراه برای پردازش و ارتباطات استفاده می‌کنند. دستگاه‌های اینترنت اشیا صنعتی از دستگاه‌های ناهمگن تشکیل شده‌اند که از طریق اینترنت به هم متصل می‌شوند و با یکدیگر ارتباط برقرار می‌کنند. استفاده گسترده از سیستم‌عامل اندروید، آن‌ها را به یک هدف ایده‌ال برای سازندگان بدافزارهای اندرویدی مخرب تبدیل کرده است. در نتیجه، مهاجمان مدل‌های بدافزار پیچیده متعددی را برای به خطر انداختن امنیت سیستم‌عامل اندروید توسعه داده‌اند [۱].

در میان گزینه‌های امنیتی مختلف، تشخیص بدافزار یک روش امنیتی پرکاربرد است که می‌تواند از منتشرشدن بدافزار در بازارهای برنامه‌های کاربردی اندروید یا از نصب و استفاده از آن‌ها در تلفن‌های همراه جلوگیری کند. استفاده گسترده از روش‌های مبهم سازی[1] کد و بارگذاری پویای کد باعث شده است روش های سنتی دیگر نتوانند عملکرد مناسبی داشته باشند. تشخیص بدافزار را می‌توان به سه دسته ایستا، پویا و ترکیبی تقسیم‌بندی کرد [۷]. به‌طورکلی در تحلیل ایستا بدون اجرای اپلیکیشن، ساختار و ویژگی‌های آن بررسی می‌شود. این امر امکان تجزیه‌وتحلیل سریع و کارآمد را فراهم می‌کند، اما ممکن است انواع خاصی از بدافزار را که رفتار مخرب خود را پنهان می‌کنند یا به اجرای پویا برای فعال‌شدن رفتارهای مخرب متکی هستند را نادیده بگیرد. تحلیل پویا شامل اجرای بدافزار در یک محیط کنترل شده و مشاهده رفتار آن است. این رویکرد می‌تواند به طور مؤثر بدافزارهایی را که به فعالیت پویا متکی هستند، شناسایی کند. البته این روش زمان‌بر و از نظر منابع پرهزینه است و ممکن است نتواند همه انواع رفتارهای مخرب را شناسایی کند. تحلیل ترکیبی، روش‌های هر دو تحلیل ایستا و پویا را برای استفاده از نقاط قوت هر رویکرد ترکیب می‌کند. به نظر رسد این روش تشخیص جامع‌تر و قوی‌تری را فراهم می‌کند [۲]؛ اما برای اثبات این موضوع باید بررسی دقیق‌تری در خصوص میزان اثربخشی و کارایی آن به عمل آورد. لازم به ذکر است در این پژوهش صرفا از تحلیل ایستا استفاده شده است.

تحقیقات نشان میدهند یادگیری ماشین یک روش مؤثر برای تشخیص بدافزار ها است. یادگیری ماشین یکی از شاخه های هوش مصنوعی است که با هر سه دسته تشخیص ایستا، پویا و ترکیبی به کار گرفته میشود. در مقایسه با روش‌های سنتی، مانند تشخیص بدافزار مبتنی بر امضا که بر شناسایی الگوهای خاص بدافزار های شناخته شده استوار است، تشخیص مبتنی بر یادگیری ماشین توانایی شناسایی انواع بدافزارهایی را دارد که قبلاً مشاهده نشده‌اند [۸]. همچنین این نوع روش‌ها عملکرد بهتری هم در زمینه کارایی[2] و هم در زمینه اثرگذاری[3] دارند [۹]. درحالی‌که که روش های سنتی نمی‌توانند به‌راحتی بدافزارهای با الگوی جدید را شناسایی کنند، روش‌های مبتنی بر یادگیری ماشین می‌توانند الگوهای رفتاری بدافزارها را یاد بگیرند و در تشخیص بدافزارهای جدید استفاده نمایند.

وجود قوانین و مقررات مانع از جمع‌آوری و اشتراک‌گذاری مستقیم داده‌های توزیع‌شده و منابع محاسباتی بین مناطق یا سازمان‌های مختلف برای انجام وظایف یادگیری ماشین می‌شود. این امر چالش‌های جدیدی را برای توسعه و پیاده‌سازی الگوریتم‌های یادگیری ماشین که نیازمند داده‌های بزرگ و محاسبات گسترده هستند ایجاد می‌کند. یادگیری ائتلافی[4] راهکاری نوین برای استفاده از داده‌ها و منابع محاسباتی پراکنده است که به دستگاه‌ها و سازمان‌های مختلف امکان می‌دهد با مشارکت جمعی، مدل‌های یادگیری ماشینی را آموزش دهند. این رویکرد به رفع چالش‌های مربوط به داده‌های منابع توزیع شده کمک می‌کند و امکان بهره‌گیری از داده‌های متنوع و پراکنده را فراهم می‌سازد. یادگیری ائتلافی به‌جای انتقال داده‌های خام به یک مکان مرکزی، داده‌ها را روی دستگاه‌های کاربران نگه می‌دارد و در همان مکان بخشی از آموزش مدل را انجام می‌دهد. این روش مشکلات مربوط به حریم خصوصی و مالکیت داده را برطرف می‌کند و به چندین گره اجازه می‌دهد تا با هم یک مدل را آموزش دهند.

به دلیل پیشرفت مداوم بدافزارهای اندرویدی، طبقه‌بندهای آموزش‌دیده برای تشخیص بدافزار، به‌مرورزمان دقت خود را از دست می‌دهند [۱۰]. به‌عبارت‌دیگر رانش مفهوم یک چالش جدی در این حوزه است. یادگیری ماشینی سنتی بر روی

[1] obfuscation

[2] Efficiency

[3] Effectiveness

[4] Federated learning

مجموعه‌داده‌هایی آموزش می‌بینند که تصویری از دنیای واقعی را در یک نقطه خاص از زمان ثبت می‌کنند. به همین دلیل، آن‌ها در برابر رانش مفهوم آسیب‌پذیر هستند. دنیای واقعی در حال تغییر است. بدافزارها دائماً در حال تکامل هستند و روش‌های جدیدی برای مخفی‌کردن فعالیت‌های خود پیدا می‌کنند. در صورت بروز رانش مفهوم، مدل‌های یادگیری ماشین آموزش‌دیده ممکن است دیگر قادر به ارائه پیش‌بینی‌های دقیق نباشند. یادگیری افزایشی، روشی در یادگیری ماشین است که می‌تواند برای مقابله با رانش مفهوم استفاده شود. یادگیری افزایشی به مدل‌ها اجازه می‌دهد تا به طور مداوم با داده‌های جدیدی که در طول زمان جمع‌آوری می‌شوند، به‌روز شوند. روش‌های متفاوتی برای استفاده از یادگیری افزایشی وجود دارد که یکی از این روش‌ها استفاده از یادگیری جمعی[1] است. یادگیری جمعی شامل سه نوع دسته‌بندی[2]، تقویتی[3] و پشته‌ای[4] است که در این پژوهش از نوع پشته‌ای استفاده شده است.

برای تشخیص بدافزار مجموعه‌داده‌های مختلفی از جمله Drebin[۱۱]، MalGenome[۱۲] و sherlock[۱۳] ارائه شده‌اند. در این پژوهش از CICMalDroid 2020 استفاده شده است. این مجموعه داده یکی از جدیدترین مجموعه‌داده‌هایی است که در دسترس عموم قرارداده شده است [۱۴].

نوآوری‌های این پژوهش:

۱. معرفی یک الگوریتم یادگیری ائتلافی - افزایشی به‌منظور بهره‌گیری از مزایای هر دو الگوریتم، مانند توزیع‌شدگی و حل مشکل رانش مفهوم

۲. استفاده از روش پشته‌ای برای یادگیری افزایشی

در این پژوهش برای تشخیص بدافزار، یک الگوریتم با استفاده از یادگیری ائتلافی و افزایشی برای بهره‌گیری از خواص هر دوی این روش‌ها پیشنهاد شده است. از جمله این خواص می‌توان به عدم نیاز به یادگیری مدل از ابتدا با همه داده‌ها در زمان رانش مفهوم، حفظ حریم خصوصی و توزیع‌شدگی اشاره کرد.

پیاده‌سازی و آزمون روش پیشنهادی روی مجموعه‌داده CICMaldroid 2020 نشان می‌دهد که این روش دقت قابل‌رقابت با بهترین روش‌های موجود برای شناسایی بدافزارها و درعین‌حال انعطاف و همچنین کارایی محاسباتی بالاتری دارد.

ادامه بخش‌های پژوهش به این شکل سازماندهی شده‌اند: در بخش دو به بررسی مفاهیم پایه مرتبط با موضوع پرداخته شده است. در بخش سه پژوهش‌های پیشین مرتبط با موضوع ارائه شده‌اند. در بخش چهار به بررسی روش‌های استفاده شده پرداخته شده است. بخش پنج شامل ارزیابی نتایج به‌دست‌آمده است. در بخش شش نتیجه‌های این پژوهش ارائه شده‌اند.

۲. مفاهیم پایه

سیستم‌عامل اندروید در حال حاضر محبوب‌ترین سیستم‌عامل برای گوشی‌های هوشمند و تبلت‌ها محسوب می‌شود. تقریباً ۲/۵ میلیارد نفر کاربر فعال از این سیستم‌عامل استفاده می‌کنند و این تعداد دائماً در حال افزایش است [۱۵]. این محبوبیت منجر به افزایش سریع نرم‌افزارهای اندرویدی شده است. در سال‌های اخیر تعداد بدافزارها هم رو به افزایش بوده است. آمار نشان می‌دهد که تنها در سال ۲۰۱۶ بیش از ۳/۲۵ میلیون برنامه اندروید که آلوده به بدافزار بودند کشف شدند. به عبارت بهتر تقریبا هر ۱۰ ثانیه یک بدافزار اندرویدی جدید شناسایی شده است [۲].

بدافزار نوعی برنامه کاربردی است که می‌تواند خدمت یا وظایف نرمال یک سامانه یا بقیه برنامه‌های کاربردی را به‌صورت عمدی مختل کند. بدافزارها از آسیب‌پذیری‌های سامانه‌های اندرویدی استفاده می‌کنند تا باعث آسیب به سامانه یا کاربر سامانه شوند. با افزایش استفاده از دستگاه‌های اندروید، پتانسیل حملات نرم‌افزارهای مخرب برای دسترسی غیرمجاز به اطلاعات افراد چندبرابر شده است. در سال‌های اخیر و به‌خصوص بعد از شیوع ویروس کرونا، وابستگی افراد به تلفن‌های همراه بیشتر شده است و افراد اطلاعات زیادی مانند اطلاعات شخصی، مالی و تحصیلی را در تلفن‌های همراه خود ذخیره کرده‌اند. پیشرفت نرم‌افزارهای مخرب اندروید با نرخ نگران‌کننده‌ای اتفاق می‌افتد و امروزه هکرها با دسترسی به ابزارهای پیشرفته، نرم‌افزارهای مخربی توسعه می‌دهند که به‌سختی قابل‌شناسایی هستند.

دسته‌بندی‌های مختلفی برای بدافزارها ارائه شده است. در [۱۶] بدافزارها به ۷ دسته تقسیم‌بندی شده‌اند. دسته اول جاسوس‌افزارها[5] هستند. یک جاسوس‌افزار نوعی بدافزار است که می‌خواهد اطلاعاتی درباره یک فرد یا سازمان را جمع‌آوری کند و سپس آن اطلاعات را به یک موجودیت دیگر ارسال کند این کار برای آسیب‌زدن به کاربر انجام می‌شود. به‌عنوان‌مثال حریم‌خصوصی کاربر نقض می‌گردد یا امنیت دستگاه او به‌خطر می‌افتد [۲].

دسته دوم کرم‌ها[6] هستند که توانایی بازتولید خود را دارند و کپی‌های خود را به دیگر رایانه‌های موجود در شبکه می‌فرستند. هدف کرم‌ها معمولاً استفاده از منابع است تا بتواند دسترسی کاربران به منابع را تأخیر بیندازد [۲].

[1] Ensemble learning
[2] Bagging
[3] Boosting
[4] Stacking
[5] Spyware
[6] Worm

دسته سوم آگهی‌افزارها[1] هستند که همانند جاسوس‌افزارها دارای اثر تخریبی نمی‌باشند و وظیفه آن‌ها بازکردن صفحات خاص اینترنتی جهت اهداف تجاری و تبلیغی است [۲].

دسته چهارم برنامه‌های کاربردی نرم‌افزاری به نام باتنت هستند که بر اساس دستورات کارهایی را به‌صورت خودکار انجام می‌دهد. این بات‌ها می‌توانند با اهداف بدخواهانه به عنوان بدافزار استفاده شوند. باتنت‌ها[2] شبکه‌ای از رایانه‌های آلوده به این نوع بدافزار هستند. این شبکه‌ها توسط یک یا چند مهاجم باهدف انجام فعالیت‌های مخرب کنترل می‌گردند. به عبارت بهتر، بات‌ها کدهای مخربی هستند که بر روی کامپیوترهای میزبان اجرا می‌شوند تا امکان کنترل نمودن آن‌ها از راه دور را برای مهاجم‌ها فراهم نمایند و آن‌ها بتوانند این مجموعه را وادار به انجام فعالیت‌های مختلف نمایند. معمولاً بات‌ها در تعداد زیاد برای ایجاد یک باتنت استفاده می‌شوند [۲].

دسته پنجم تروجان‌ها[3] هستند. تروجان یک برنامه مضر است که خود را به‌عنوان یک برنامه معمولی و خوش‌خیم به‌منظور ترغیب یک قربانی برای نصب آن سوق می‌دهد. یک تروجان معمولاً عملکرد مخرب مخفی را انجام می‌دهد که هنگام شروع برنامه فعال می‌شود [۲].

دسته ششم درب‌های پشتی[4] هستند. درب پشتی برنامه‌ای هست که به نفوذگر این امکان را می‌دهد تا با دورزدن روند امنیتی سامانه، منابع مختلفی از آن سامانه را از راه مربوطه در اختیار نفوذگر قرار بدهد [۲].

دسته آخر باج‌افزارها[5] هستند که سامانه را آلوده و سپس برای بازگرداندن آن به حالت عادی درخواست پول می‌کنند. مثلاً بعضی از فایل‌ها را در سامانه رمزنگاری می‌کنند و برای رمزگشایی آن درخواست پول می‌نمایند [۲].

بعضی از محققان برای تشخیص انواع این بدافزارها از یادگیری ماشین استفاده نموده‌اند. یکی از چالش‌های این محققان پدیده‌ای به نام رانش مفهوم[6] است [۱۷]. این پدیده به تغییر رابطه بین متغیرهای ورودی و خروجی مدل در طول زمان گفته می‌شود. وقوع این پدیده باعث می‌گردد مدل آموزش‌دیده با گذشت زمان دقت خود را از دست بدهد. برای اجتناب از وقوع چنین مشکلی می‌توان از یادگیری افزایشی بهره برد. این نوع یادگیری مزایای قابل‌توجهی نسبت به یادگیری سنتی دارد. یادگیری افزایشی به مدل کمک می‌کند تا دانش خود را در مورد دنیای واقعی به‌روز کند و در نتیجه در برابر رانش مفهوم مقاوم باشد. همچنین می‌تواند نیاز به حافظه و محاسبات را کاهش دهد و امکان تصمیم‌گیری بلادرنگ را فراهم نماید.

یکی از روش‌های مناسب برای یادگیری افزایشی روشی به نام روش پشته‌ای است[۱۸]. در این روش، مجموعه داده‌ها به چند قسمت تقسیم می‌گردد. مدل اولیه بر روی هر قسمت بصورت جداگانه آموزش می‌بینند. سپس پیش‌بینی‌های حاصل از مدل‌های ایجاد شده به عنوان داده ورودی مورد استفاده قرار می گیرند و مدل‌های جدیدی بر اساس این داده‌ها ایجاد می‌گردند. این روند تا حصول مدل مناسب تکرار می‌گردد. روش پشته‌ای می‌تواند برای افزایش دقت پیش‌بینی در طیف گسترده‌ای از مسائل یادگیری ماشین استفاده شود. این تکنیک به‌ویژه برای مسائلی که در آن داده‌ها ناقص یا دارای نویز هستند نیز مفید است. در ادامه پژوهش‌های انجام شده مرتبط با موضوع این پژوهش بررسی گردیده است.

۳. پژوهش‌های پیشین

ممون و همکاران [۱۹] از توزیع‌شدگی مبتنی بر اسپارک برای افزایش سرعت تشخیص بدافزار استفاده کرده‌اند. آنان با ایجاد یک خوشه ۱۷ گره‌ای در اسپارک و استفاده از بخشی از مجموعه داده SherLock الگوریتم‌های مختلفی از یادگیری ماشین را تست کرده‌اند. در این میان بالاترین دقت بدست آمده مربوط به مدل درختان تقویت شده با گرادیان ۱۷ بوده است. این مدل توانسته است با دقت تقریبا ۹۰درصد بدافزار را تشخیص دهد. در این الگوریتم از یادگیری افزایشی استفاده نشده است و در صورت وجود داده جدید باید مرحله آموزش دوباره از ابتدا اجرا شود. به عبارت دیگر روش مورد نظر نسبت به رانش مفهوم آسیب‌پذیر است. با توجه به توسعه سریع بدافزارهای اندرویدی جدید، تکرار زیاد مرحله آموزش باعث ناکارآمدشدن استفاده از چنین روشی می‌گردد.

کومار [۲۰] مدل‌های یادگیری ماشین را با استفاده از اسپارک به‌صورت توزیع‌شده پیاده‌سازی کرده است. ویژگی توزیع‌شدگی باعث کاهش زمان تشخیص و یادگیری شده است. در بخش ارزیابی علاوه بر معیار صحت، معیار میزان زمان مورد نیاز برای یادگیری و تشخیص نیز اندازه‌گیری شده است. در این روش از یادگیری افزایشی استفاده نشده و به نظر می‌رسد نسبت به رانش مفهوم نیز آسیب‌پذیر است.

جوشی و کومار [۲۱] از روش پشته‌ای برای تشخیص بدافزارهای اندرویدی استفاده کرده‌اند. این روش از چهار مدل یادگیری ماشین مختلف استفاده می‌کند تا دقت تشخیص بدافزار را بهبود

[1] Adware
[2] Botnet
[3] trojan
[4] Backdoor
[5] Ransomware
[6] Concept drift

ببخشد. نتیجه تست‌ها نشان می‌دهد که دقت به دست آمده از اجرای این روش بر روی مجموعه داده CIC-MalDroid 2020، ۹۸ درصد و بر روی مجموعه داده CIC-MalMem 2022، ۹۹,۹۹درصد است. این روش تشخیص خود را مبتنی بر یادگیری ائتلافی یا افزایشی انجام نمی‌دهد و توزیع‌شده نیز نمی‌باشد. همچنین نسبت به رانش مفهوم آسیب‌پذیر است. از طرف دیگر در این پژوهش معیار سرعت تشخیص بدافزار مورد ارزیابی قرار نگرفته است.

جنت و همکاران [۲۲] سامانه‌ای مبتنی بر یادگیری ماشین برای تحلیل و شناسایی بدافزارها در اندروید پیشنهاد کرده‌اند. این سامانه با دو روش تحلیل پویا و تحلیل ایستا به بررسی بدافزارها می‌پردازد. نویسندگان در تحلیل ایستا از دو مجموعه‌داده استفاده کرده‌اند. مجموعه داده MalGenome که شامل حدود ۱۲۰۰ برنامه بدافزار طبقه‌بندی‌شده بر اساس خانواده‌های آن‌هاست. مجموعه داده دوم[۲۳] نیز یکی از مجموعه‌های موجود در وبسایت Kaggle است که شامل اطلاعات ۴۰۰۰ برنامه مخرب می‌باشد. از طرف دیگر برای تحلیل پویا، از الگوریتم جنگل تصادفی و مجموعه‌داده MalGenome استفاده شده و دقت ۹۳ درصد به دست آمده است. در این پژوهش از یادگیری ائتلافی یا افزایشی استفاده نشده است و نسبت به رانش مفهوم نیز آسیب‌پذیر است.

چن و همکاران [۲۴] یک روش جدید مبتنی بر یادگیری عمیق برای تشخیص بدافزار اندرویدی ارائه کرده‌اند. این روش از اطلاعات موجود در بسته‌های برنامه اندروید، مانند مجوزها، خدمات، گیرندگان و قصدها، برای استخراج ویژگی‌های برنامه استفاده می‌کند. سپس این ویژگی‌ها به یک مدل BiLSTM [۲۵] داده می‌شوند تا برچسب برنامه بی‌خطر یا بدخواه را پیش‌بینی کنند. در ارزیابی انجام شده بر روی مجموعه داده CICMalDroid 2020، این روش به دقت ۹۷/۴۷ درصد دست یافته است. در این پژوهش از یادگیری ائتلافی یا افزایشی استفاده نشده است.

جواهری و همکاران [۲۶] یک روش بهینه برای شناسایی باج‌افزارها بر اساس رفتارهای آن‌ها با استفاده از توابع سامانه‌ای ارائه کرده‌اند. این روش با تولید یک مجموعه داده غنی شامل انواع خانواده‌ها و نسخه‌های مختلف باج‌افزارها آغاز شده است. از مجموعه داده اولیه با ۱۲۶ ویژگی، با اعمال چهار مرحله ارتقاء و بازنگری، مجموعه داده به ۶۷ ویژگی بهینه شده است. این پژوهش از الگوریتم دسته‌بندی جنگل تصادفی استفاده کرده است و با انجام بهینه‌سازی‌هایی مثل بهینه‌سازی مجموعه داده‌ها اقدام به ارتقاء نرخ سرعت تشخیص و همچنین حفظ دقت کرده است. در این روش تنها باج‌افزارها قابل شناسایی هستند. همچنین در این پژوهش از یادگیری افزایشی یا توزیع‌شدگی استفاده نشده است. به نظر می‌رسد این روش نیز نسبت به رانش مفهوم آسیب‌پذیر است.

حسابی و همکارش [۲۷] یک روش نوین برای تشخیص بدافزارها و حملات در محیط رایانش ابری ارائه کرده‌اند. این روش از خوشه‌بندی داده‌ها برای جداسازی آن‌ها از یکدیگر و متوازن‌سازی کلاس‌های مختلف استفاده می‌کند تا شرایط مناسبی برای ساخت مدل فراهم شود. این پژوهش از تکنیک‌های یادگیری جمعی و ساخت مدل سطح بالا با استفاده از مکانیزم رأی‌گیری نیز بهره می‌گیرد. در مدل پیشنهادی این پژوهش، یادگیری جمعی با استفاده از نقاط قوت الگوریتم‌های مختلف مانند الگوریتم جنگل تصادفی و الگوریتم آدابوست، یک سامانه با عملکرد بالا را برای شناسایی بدافزار در رایانش ابری ایجاد می‌نماید. در این روش از یادگیری ائتلافی یا افزایشی استفاده نشده و نسبت به رانش مفهوم آسیب‌پذیر است.

عزت نشان و همکاران [۲۸] رویکردی را برای ارزیابی و پیش‌بینی حملات باتنت‌ها ارائه کرده‌اند. در پژوهش آن‌ها بیان شده است که علیرغم اینکه تاکنون تحقیقات زیادی در زمینه تشخیص باتنت‌ها انجام شده است، اما روش‌های موجود نتوانسته‌اند دقت بالایی را در تشخیص بدافزار داشته باشند. باتنت‌هایی که در شرایط خاص رفتار اصلی خود را بروز می‌دهند به سختی قابل تشخیص هستند. همچنین این روش‌ها باید برای مقایسه با گذشته، تاریخچه کامل باتنت‌ها را به‌یاد بسپارندکه این امر در عمل غیرممکن است. البته به منظور کاهش نیاز به حافظه یک ساختار مبتنی بر زنجیره مارکوف ارائه شده است. این رویکرد بر اساس تحلیل رفتاری باتنت‌ها می‌باشد و بدون نیاز به حافظه‌ای بزرگ عمل می‌کند. در این روش تنها روی شناسایی باتنت‌ها تمرکز شده و به شناسایی انواع دیگر بدافزارها پرداخته نشده است. همچنین در این پژوهش از یادگیری افزایشی یا توزیع‌شدگی استفاده نشده و نسبت به رانش مفهوم آسیب‌پذیر است.

مصلح و همکارش [۲۹] یک رویکرد جدید را با استفاده از هم‌افزایی ویژگی‌های شمارنده‌های سخت‌افزاری و شبکه عصبی پرسپترون چندلایه برای تشخیص بدافزارها با استفاده از هوش مصنوعی و داده‌کاوی ارائه کرده‌اند. سامانه پیشنهادی با استخراج ویژگی‌های با تفکیک‌پذیری بالا و استفاده از شبکه عصبی بهینه‌شده، توانسته است به خوبی فایل‌های سالم را از فایل‌های مخرب تشخیص دهد. همچنین استفاده از الگوریتم سنجاقک در بهینه‌سازی شبکه عصبی، منجر به عملکرد بهتری شده و سامانه پیشنهادی با دقت ۸۶ درصد، فایل‌های آلوده به بدافزار را تشخیص داده است. در این روش از یادگیری ائتلافی یا افزایشی

استفاده نشده و نسبت به رانش مفهوم آسیب‌پذیر است.

هسو و همکاران [۳۰] یک سامانه یادگیری ائتلافی حفظ‌کننده حریم خصوصی (PPFL) را برای تشخیص بدافزار اندرویدی ارائه کرده‌اند. سامانه PPFL به دستگاه‌های تلفن همراه اجازه می‌دهد تا با هم برای آموزش یک طبقه‌بند همکاری کنند و در همان حال اطلاعات و عملیات‌های حساس خود مانند فراخوانی‌های API و تنظیمات مجوز را از هر گونه دسترسی غیرمجاز محفوظ نگه دارند. در واقع، هر دستگاه تلفن همراه مدل خود را روی داده‌های خود آموزش می‌دهد و سپس این مدل‌های محلی با استفاده از تکنیک‌های محاسبات امن چندجانبه ترکیب می‌شوند تا یک مدل سراسری ایجاد شود. این مدل سراسری برای تشخیص بدافزار استفاده می‌شود. این کار سامانه یادگیری ائتلافی حفظ‌کننده حریم خصوصی را بر اساس ماشین بردار پشتیبان (SVM) و تکنیک‌های محاسبات امن چندجانبه پیاده‌سازی می‌کند. نتایج آزمایش روش پیشنهادی بر روی مجموعه داده‌های بدافزارهای اندرویدی تهیه شده توسط موسسه ملی فناوری اطلاعات و ارتباطات ژاپن نشان می‌دهد که عملکرد طبقه‌بند PPFL از عملکرد سامانه آموزش متمرکز پیشی می‌گیرد. علاوه بر این، حریم خصوصی اطلاعات برنامه شامل اطلاعات API، مجوزها و مدل‌های محلی آموزش‌دیده تضمین می‌شود. بنا به ادعای این مقاله، این کار اولین سامانه تشخیص بدافزار اندرویدی است که بر اساس سامانه یادگیری ائتلافی حفظ کننده حریم خصوصی ساخته شده است. در این روش از یادگیری افزایشی استفاده نشده است و نسبت به رانش مفهوم آسیب‌پذیر است.

گالوز و همکاران [۳۱] یک چارچوب طبقه‌بندی بدافزار به نام LiM را ارائه می‌دهند که از یادگیری ائتلافی برای تشخیص و طبقه‌بندی برنامه‌های مخرب با رعایت حریم خصوصی کاربران استفاده می‌کند. LiM اطلاعات مربوط به برنامه‌های تازه نصب‌شده را به صورت محلی در دستگاه‌های کاربران نگه می‌دارد تا از استنباط برنامه‌های نصب‌شده کاربران توسط کارگزار جلوگیری کند. این چارچوب از ورودی همه کاربران استفاده می‌نماید و حریم خصوصی کاربران را نیز حفظ می‌کند. همچنین عملکرد طبقه‌بندی را بهبود می‌بخشد. برای رفع چالش عدم دسترسی کاربران به اطلاعات درست پایه‌ای[1]، LiM از یک مجموعه نیمه‌نظارت‌شده امن استفاده می‌کند که دقت طبقه‌بندی را نسبت به یک طبقه‌بندی‌کننده پایه مبتنی بر ابر، افزایش می‌دهد. در این پژوهش از یادگیری افزایشی استفاده نشده است و نسبت به رانش مفهوم آسیب‌پذیر است.

لک و همکاران [۳۲] با استفاده از شبکه عصبی چندلایه و الگوریتم بهینه‌سازی مبتنی بر آموزش و یادگیری، توانست با دقت ۹۹٪ و صحت ۹۸٪ به نتایج بسیار خوبی دست یابد. نتایج نشان داد که این روش در مقایسه با الگوریتم‌های ماشین بردار، الگوریتم ژنتیک و نزدیک‌ترین همسایه، بهبود قابل توجهی داشته و در تشخیص بدافزارهای سیستم عامل اندروید عملکرد بهتری دارد. این مطالعه بر روی مجموعه‌داده جامع و عظیم بدافزار اندرویدی که توسط موسسه امنیت سایبری کانادا ارائه شده بود، انجام شده است.

قنواتی نسب و همکاران [۳۳] برای تشخیص باتنت‌ها، با استفاده از مهندسی معکوس، لیست مجوزهای برنامه‌ها را استخراج و سپس بر اساس این لیست، تصاویر معادلی از برنامه‌ها ایجاد می‌کنند. این تصاویر با استفاده از شبکه عصبی کانولوشنال طبقه‌بندی شده و نوع برنامه کاربردی تعیین می‌شود. نتایج این روش نشان داده که عملکرد بهتری نسبت به روش‌های سنتی یادگیری ماشین نظیر ماشین بردار پشتیبان و درخت تصمیم دارد.

بسطامی و همکاران [۳۴] روشی را پیشنهاد دادند که از تعبیه‌سازی دنباله‌های کد دستوری بدافزارها و تبدیل آن‌ها به تصاویر استفاده می‌کند. این روش با استفاده از مدل Word2Vec، دنباله‌های کد عملیاتی را به تصاویر سیاه و سفید تبدیل کرده و سپس با بهره‌گیری از شبکه عصبی کانولوشنی، تصاویر را آموزش داده و بدافزارها را شناسایی و طبقه‌بندی می‌کند. نتایج این پژوهش نشان داده که دقت طبقه‌بندی با استفاده از این روش به حداکثر ۱۰۰ درصد و میانگین ۸۹,۹۸ درصد می‌رسد که بالاتر از سایر الگوریتم‌های موجود است. با این حال، این روش در صورت کم بودن تعداد نمونه‌ها و یا استفاده بدافزار از تکنیک‌های مخفی‌سازی عملکرد ضعیف‌تری دارد.

۴. روش پیشنهادی

مدل استفاده شده در این پژوهش پرسپترون چندلایه (MLP) با یک‌لایه ورودی، یک‌لایه مخفی با ۲۰۰ نورون و یک‌لایه خروجی است. تعداد نورون‌های لایه مخفی بعد آزمون چند عدد و مقایسه دقت‌ها انتخاب شده است.

۴-۱. پیش‌پردازش و تقسیم داده‌ها بین گره‌ها

در این پژوهش گره‌ها می‌توانند تلفن‌های همراه یا دستگاه‌های IoT باشند. در این پژوهش ابتدا در مجموعه‌داده به‌جای مقادیر خالی صفر قرار داده می‌شود [۳۵]. سپس داده ها به صورت تصادفی به سه بخش آموزش، تست و اعتبارسنجی تقسیم می شوند. دو بخش تست و اعتبارسنجی در اختیار کارگزار قرار می گیرند و بخش آموزش بین گره‌ها تقسیم می‌شود. همچنین یک

[1] Ground truth

وزن تصادفی هم از سمت کارگزار تولید شده و به همه گره‌ها ارسال می‌شود. گره‌ها بعدا برای آموزش مدل‌های خود از این وزن شروع می‌کنند.

۴-۲. انتخاب ویژگی

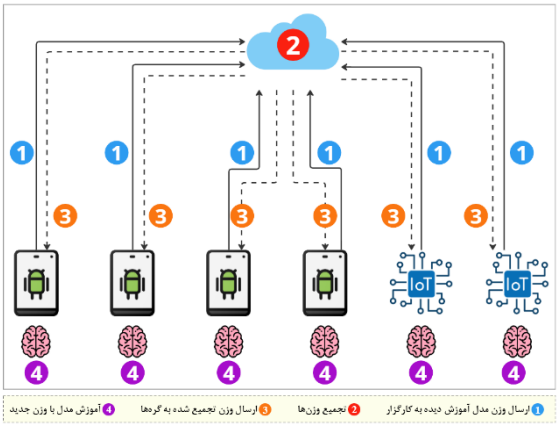
باتوجه‌به این که در مدل‌های یادگیری ائتلافی وزن‌ها بین گره‌ها و کارگزار جابه‌جا می‌شوند و اندازه وزن مدل به تعداد ویژگی‌ها وابسته است، برای کاهش بار شبکه، تعداد ویژگی‌ها کم شده است. انتخاب ویژگی فرایندی است که در آن ویژگی‌های اضافی یا غیرمرتبط از مجموعه‌داده اصلی حذف می‌شوند. این کار دو مزیت عمده دارد: با حذف ویژگی‌های غیرضروری، حجم داده‌هایی که طبقه‌بند باید پردازش کند کاهش می‌یابد. این امر منجر به اجرای سریع‌تر و کارآمدتر مدل می‌شود. همچنین ویژگی‌های غیرمرتبط می‌توانند بر دقت طبقه‌بندی تأثیر منفی بگذارند. حذف این ویژگی‌ها به مدل اجازه می‌دهد تا روی ویژگی‌های مرتبط تمرکز کند و در نتیجه، دقت کلی مدل را افزایش دهد [۳۶].

در این پژوهش از روش آستانه واریانس[1] برای حذف ویژگی‌های کم‌اهمیت استفاده شده است. این روش تمام ویژگی‌هایی را که واریانس آن‌ها کمتر از یک مقدار آستانه باشد حذف می‌کند. به طور پیش‌فرض، تمام ویژگی‌هایی که واریانس آن‌ها صفر است، یعنی ویژگی‌هایی که در تمام نمونه‌ها یک مقدار یکسان دارند، حذف می‌شوند. علاوه بر این در این پژوهش برای تمرکز روی ویژگی‌های دارای تنوع و اطلاعات مناسب، حداقل آستانه واریانس ۰/۰۰۰۱ در نظر گرفته شده است. باتوجه‌به این آستانه، تعداد ویژگی‌های انتخابی از ۵۰۶۲۱ به ۵۷۵ ویژگی به‌صورت میانگین کاهش می‌یابد.

در این پژوهش انتخاب ویژگی‌ها روی مجموعه‌داده اعتبارسنجی انجام می‌گردد. سپس ویژگی‌های انتخابی به گره‌ها هم ارسال می‌شوند تا آن‌ها نیز همان مجموعه از ویژگی‌ها را انتخاب کنند. دلیل این نوع شیوه انتخاب این است که دسترسی لازم به داده‌های گره‌ها در طرف کارگزار وجود ندارد و نمی‌توان از ابتدا انتخاب ویژگی را روی همه داده‌ها انجام داد.

۴-۳. یادگیری ائتلافی

در بخش بعدی جزئیات روش پیشنهادی ارائه خواهد شد. به‌منظور درک بهتر روش مذکور، در این بخش فرایند یادگیری ائتلافی موردبررسی قرار گرفته است. نمونه‌ای از معماری و فرایند یادگیری ائتلافی در شکل (۱) دیده می‌شود. گره‌ها با وزن‌های اولیه‌ای که از کارگزار دریافت می‌کنند فرایند آموزش را روی مجموعه‌داده خود انجام می‌دهند. در این معماری برای جلوگیری از تحمیل بار پردازشی زیاد به گره‌ها، بیشترین تعداد چرخش مدل یادگیر برابر ۱ قرار داده شده است. این موضوع در شکل نشان داده نشده است.



شکل (۱). نمایش گرافیکی یادگیری ائتلافی

پس از آموزش، گره‌ها وزن‌های یادگیری شده را به کارگزار می‌فرستند. کارگزار، میانگین وزن‌دار وزن‌های دریافتی را محاسبه می‌کند. سپس این میانگین به گره‌ها ارسال می‌شود. در سمت گره‌ها دوباره یادگیری روی داده‌های محلی با استفاده میانگین رسیده انجام می‌شود و وزن‌های یادگیری شده به سمت کارگزار ارسال می‌شوند. این کار باید چندین بار تکرار شود که در این پژوهش این عمل بیست بار انجام شده است. همچنین برای افزایش دقت از یک زمان‌بند نرخ یادگیری[2] در مدل پیشنهادی استفاده شده است. یک نرخ یادگیری ثابت که در ابتدای آموزش مؤثر است، ممکن است در ادامه به دلیل نوسانات زیاد یا عدم توانایی در رسیدن به کمینه محلی، کارایی خود را از دست بدهد. زمان‌بند نرخ یادگیری با کاهش تدریجی نرخ یادگیری در طول زمان، این مشکل را حل می‌کند و به مدل اجازه می‌دهد تا به‌آرامی به یک کمینه پایدار نزدیک شود. در هر بار تکرار نرخ یادگیری تقسیم بر ۱/۳(یک و سه دهم) می‌شود. این عدد با آزمودن بیش از ۲۰ عدد مختلف انتخاب شده است.

نکته قابل‌توجه این است که میانگین‌گیری در این پژوهش به‌صورت وزن‌دار انجام می‌شود. هر بار که وزن‌ها به کارگزار می‌رسند، دقت مدلی که با این وزن‌ها روی داده اعتبارسنجی پیش‌بینی انجام می‌دهد سنجیده می‌شود. این دقت‌ها به یک تابع

[1] VarianceThreshold

[2] Learning rate schedule

Softmax داده می‌شوند تا آن‌ها را به‌گونه‌ای تغییر دهد که مجموع آن‌ها یک شود. فرمول محاسبه تابع Softmax برای بردار $\{x_0,x_1,\ldots x_{n-1}\}$ به‌صورت رابطه زیر است:

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (۱)$$

تابع Softmax برداری از اعداد را به برداری از احتمالات تبدیل می‌کند. احتمال محاسبه‌شده برای هر مقدار متناسب با مقیاس نسبی آن مقدار در بردار است. به‌عبارت‌دیگر، هرچه یک عدد در بردار بزرگ‌تر باشد، احتمال آن نیز بیشتر است. باتوجه‌به اینکه مجموع احتمالات به‌دست‌آمده یک است، لذا احتمالات محاسبه شده در تعداد گره‌ها ضرب می‌شوند تا اعداد به‌دست‌آمده نزدیک یک باشند تا در مراحل بعدی که در وزن گره ضرب می‌شوند باعث انحراف زیادی از مقدار واقعی آن‌ها نشوند. سپس مقدار به‌دست‌آمده برای هر گره در وزن آن گره ضرب می‌شود تا گرهی که دقت بیشتری را به دست آورده است سهم بیشتری در محاسبه میانگین داشته باشد.

۴-۴. یادگیری ائتلافی - افزایشی

روش پیشنهادی در این مقاله یک الگوریتم ائتلافی - افزایشی است که ترکیبی از یادگیری ائتلافی با یادگیری افزایشی است. ابتدا با استفاده از داده‌های موجود و بهره‌گیری از روش یادگیری ائتلافی مدل مناسب ساخته می‌شود. سپس با ورود داده جدید با کمک یادگیری افزایشی مدل بروز می‌گردد. در این میان دو حالت متصور است. در حالت اول اضافه‌شدن گره جدید منجر به مشاهده داده تازه می‌شود و در حالت دوم داده جدید در یکی از گره‌های قدیمی مستقر است و همچنان برای یادگیری استفاده نشده است. در هر دوی این حالت‌ها، مدل جدیدی با اجرای یادگیری ائتلافی بر روی داده‌های جدید ایجاد می‌شود. سپس یادگیری افزایشی مدل قبلی ساخته شده را با استفاده از مدل جدید بروز می‌نماید. لازم به ذکر است در این روش برای یادگیری افزایشی از الگوریتم پشته‌ای بهره گرفته شده است.

به طور دقیق‌تر در یادگیری افزایشی دودسته طبقه‌بند پایه و طبقه‌بند کلی[1] نقش اساسی در یادگیری ایفا می‌نمایند. در روش پیشنهادی طبقه‌بندهای ایجاد شده در مرحله یادگیری ائتلافی به‌عنوان طبقه‌بندهای پایهٔ یادگیری افزایشی مورداستفاده قرار می‌گیرد. به‌عبارت‌دیگر طبقه‌بندهای پایه همان مدل‌های درون گره‌ها هستند. نحوه ایجاد این مدل‌ها در بخش قبلی توضیح داده شد. البته لازم به ذکر است با ورود داده جدید چنین مدلی نیز بر اساس این داده‌ها ایجاد می‌گردد.

پس از این‌که فرایند یادگیری ائتلافی برای گره‌های جدید پایان یافت، آخرین وزن‌های گره‌های دارای داده جدید به آخرین وزن‌های موجود به داده‌های قبلی متصل می‌شود و یک بردار جدید را به وجود می‌آورد. سپس این بردار به گره‌ها ارسال می‌شود. هر گره داده‌های خود را در این بردار ضرب می‌نماید. بردار حاصل به یک تابع لجستیک استاندارد داده می‌شود و نتیجه حاصل بردار جدیدی خواهد بود که برای آموزش طبقه‌بند کلی مورداستفاده قرار خواهد گرفت. فرمول محاسبه تابع لجستیک استاندارد به‌صورت رابطه زیر است:

$$logistic(x) = \frac{1}{1+e^{-x}} \quad (۲)$$

این بردارها به همراه برچسب واقعی خود به کارگزار ارسال می‌شوند و آموزش طبقه‌بند کلی صورت می‌پذیرد. با استفاده از این طبقه‌بند می‌توان بدافزار بودن یا نبودن یک نرم‌افزار اندرویدی را تشخیص داد. همچنین در صورت بدافزار بودن نرم‌افزار مذکور نوع آن نیز مشخص می‌گردد.

شبه‌کد روش پیشنهادی در شکل (۲) نشان‌داده‌شده است. همان‌طور که در تصویر مشخص است این روش به‌صورت کلی شامل مراحل پیکربندی اولیه الگوریتم، انجام یادگیری ائتلافی در روش پیشنهادی بر روی داده‌های موجود، پایان یادگیری ائتلافی اولیه، انجام یادگیری تکمیلی ائتلافی برای داده‌های جدید، پایان یادگیری ائتلافی داده‌های جدید و شروع یادگیری افزایشی و انجام یادگیری افزایشی است. شکل (۳) نیز نمایش گرافیکی دو مرحله آخر این فرآیند را نشان می‌دهد.

[1] Meta

#Initial algorithm configuration
1. Create server and clients
2. Divide train data among clients and placing validation and test data in server
3. Preprocessing
4. Feature selection
5. Make random weight and send it to clients

#Federated learning on existing data
1. Repeat
    1.1. Train the nodes with the weights received from the server.
    1.2. Send trained weight of models to server
    1.3. Calculate weighted average in server
    1.4. Broadcast averaged weight to clients
2. Until number_of_iterations=predefined_number_of_iterations

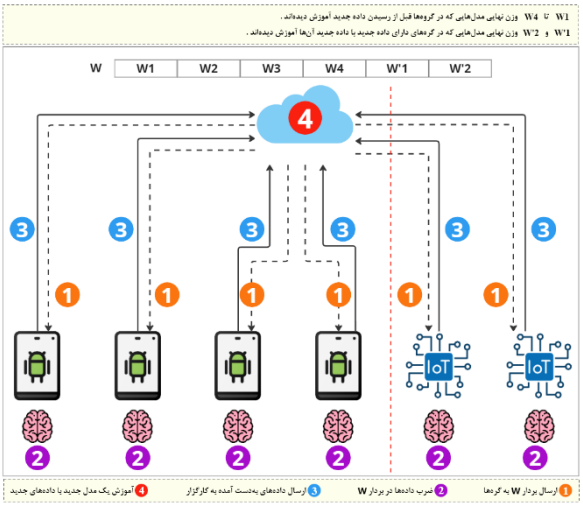
Completion of the initial federated learning phase
1. Save final weights of clients on server
2. Make model in server

#Federated learning on new data
1. Repeat
    1.1. Train with new data on the nodes containing them with the weights received from the server
    1.2. Send trained weight of models to server
    1.3. Calculate weighted average in server
    1.4. Broadcast averaged weight to clients
2. until number_of_iterations=predefined_number_of_iterations

#Completion of federated learning on new data and start incremental learning
1. Save final weights of clients on server
2. Append calculated weights for the nodes with new data to the final weights of other nodes

#Incremental learning
1. n-wise calcucate average of final weights
2. Broadcast averaged wights in previous step to clients
3. Multiply each line of data by received weights to make new data
4. Send new data to server
5. Multipy test data by weights calculated in step 2 to make new test data
6. Train final model in server

شکل(۲). شبه کد روش یادگیری افزایشی-ائتلافی



شکل (۳). نمایش گرافیکی فرآیند یادگیری افزایشی در روش پیشنهادی

برای کاهش میزان بار ارسالی در شبکه می‌توان وزن‌ها را با هم ترکیب کرد؛ مثلاً در این پژوهش با ۱۰ گره وزن‌ها دوبه‌دو با هم میانگین گرفته‌شده و ارسال شدند و کاهش دقت را نیز در برنداشت. برای تعداد ترکیب وزن‌ها می‌توان تعداد گره‌ها را تقسیم بر ۱۰ کرد و خارج‌قسمت را به‌اضافه یک نمود. برای مثال از ۱ تا ۹ گره عدد مطلوب ۱ است و نیازی به ترکیب نیست. برای ۱۰ تا ۱۹ عدد ۲ عدد مناسب است و باید ۲ تا ۲ تا وزن‌ها را با هم ترکیب کرد. این روش ترکیب در عمل نتیجه خوبی را به همراه دارد.

۴-۵. جزئیات پیاده‌سازی

روش پیشنهادی بر روی کامپیوتری که دارای پردازنده Intel Core i7-7500U ۳/۵ گیگاهرتز، حافظه اصلی ۱۲ گیگابایت و پردازنده گرافیکی NVIDIA GeForce 920MX با ۲ گیگابایت حافظه اصلی اجرا گردیده است.

همچنین سیستم‌عامل مورداستفاده نیز ویندوز ۱۱ نسخه ۶۴ بیتی بوده است. مدل پیشنهادی با استفاده از زبان برنامه‌نویسی پایتون نسخه ۳/۱۱ و در محیط ژوپیتر نوت‌بوک آموزش‌داده‌شده است.

۵. ارزیابی و نتایج

در این پژوهش از مجموعه‌داده CICMalDroid 2020 برای آموزش و تست روش پیشنهادی استفاده شده است. این مجموعه‌داده شامل ۱۱۵۹۸ نمونه با ۵۱۲۲۹ ویژگی است. این ویژگی‌ها متشکل از ۵۰۶۲۱ ویژگی شامل اطلاعات ایستا مثل مجوزها، سرویس‌ها، رابط‌های برنامه‌نویسی کاربردی[1] حساس و پکیج‌ها است که در این پژوهش مدنظر قرار گرفته‌اند. بقیه ویژگی‌ها در این مجموعه‌داده غیرایستا هستند.[۳۷, ۳۸]

همان‌طور که در بخش ۴-۴ توضیح داده شد، دو حالت برای داده جدید متصور است. به‌منظور ارزیابی روش پیشنهادی در حالت اول که داده‌های جدید در هیچ یک از گره‌های موجود مستقر نمی‌باشند، ۸ گره در نظر گرفته شده و بر روی هر گره یک‌دهم حجم کل داده‌ها قرار داده می‌شود. باقی‌مانده داده‌ها نیز بر روی دو گره جدید به‌صورت مساوی تقسیم می‌شوند تا بتوان برای ارزیابی حالت اول استفاده نمود.

در حالت دوم نیز که داده‌های جدید در گره‌های موجود مستقر هستند، ۸ گره برای آموزش در نظر گرفته شده است. بر روی هر یک از این گره‌ها یک دهم حجم کل داده‌ها قرار می گیرند. از طرف دیگر به‌منظور ارزیابی حالت دوم، دو گره از این هشت گره انتخاب شده و بر روی هر یک، نیمی از باقی‌مانده داده‌ها قرار می گیرد. به عبارت دیگر دو گره از این هشت گره دو برابر سایر گره‌ها دارای داده هستند که نیمی از آن داده‌ها در مرحله یادگیری ائتلافی اولیه و نیمی دیگر در مرحله یادگیری ائتلافی و یادگیری افزایشی با داده‌های جدید مورد استفاده قرار خواهند گرفت.

به‌منظور ارزیابی بهتر روش پیشنهادی، دو روش دیگر نیز در نظر گرفته شده است. یکی از این روش‌ها، روش متمرکز است که بدون قابلیت یادگیری ائتلافی یا افزایشی و صرفاً به‌صورت روش‌های کلاسیک آموزش داده می‌شود. روش دوم نیز یادگیری ائتلافی به‌صورت توزیع شده و بدون حضور یادگیری افزایشی است. هر یک از این روش‌ها ۱۰ بار اجرا گردیده و برای هر یک معیارهای صحت[2]، دقت[3]، امتیاز F1[4] و بازخوانی[5] از روی ماتریس درهم‌ریختگی[6] محاسبه شده است[۳۹]. نتیجه حاصل میانگین مقدار این معیارها در ۱۰بار اجرا می‌باشد که در جدول (۱) نشان داده شده است. لازم به ذکر است برای حفظ یکنواختی در شکل داده‌ها، در ارزیابی روش پیشنهادی داده‌های تست نیز همانند داده‌های آموزش در وزن‌های مدل‌ها ضرب شده و مقادیر جدید جایگزین آن‌ها گردیده است.

همچنان که در جدول (۱) مشاهده می‌شود، زمان یادگیری در روش پیشنهادی نسبت به سایر روش‌ها بیشتر می‌باشد. دلیل این امر این است که این زمان‌ها روی یک رایانه و به‌صورت غیر توزیع شده آزمایش شده‌اند؛ اما در یک مدل واقعی بخش زیادی از این

[1] Application programming interface

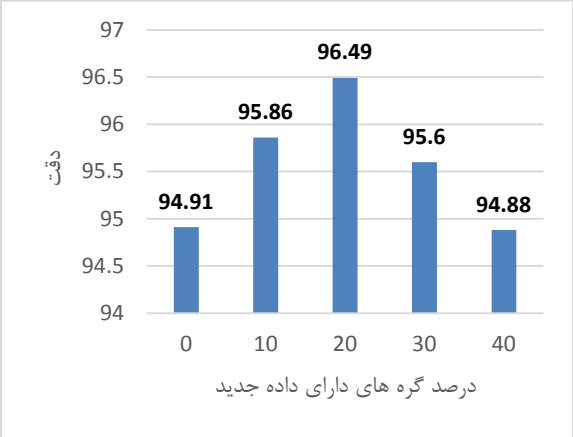
[2] Precision
[3] Accuracy
[4] F1_score
[5] Recall
[6] Confusion matrix

زمان بین گره‌ها تقسیم می‌شود. همچنین در محاسبه زمان‌ها تاخیر شبکه در نظر گرفته نشده است.

جدول (۱). مقایسه عملکرد روش پیشنهادی با مدل متمرکز و یادگیری ائتلافی

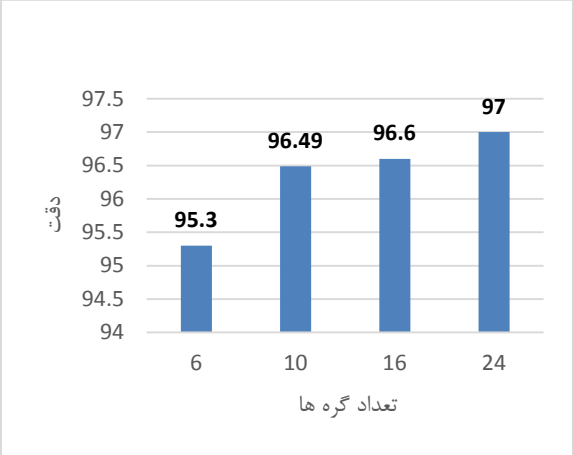
	دقت	صحت	بازخوانی	امتیاز F1	زمان
مدل متمرکز	۹۱/۲	۹۱/۲	۹۱/۲	۹۲/۴	۱۵s
یادگیری ائتلافی	۹۲/۲	۹۲/۲۱	۹۲/۲	۹۲/۴	۱۵s
یادگیری ائتلافی - افزایشی	۹۶/۴۹	۹۶/۶	۹۶/۴۹	۹۶/۴۹	۹۰s

طبق جدول (۱) روش پیشنهادی از نظر دقت، بهبود خوبی را ایجاد نموده است. این روش حتی می‌تواند بدون داده جدید هم برای افزایش دقت مورد استفاده قرار گیرد؛ یعنی داده گروهی از گره‌ها به‌عنوان داده جدید فرض شود. حجم این گروه‌ها معمولا به صورت نسبت یا درصدی از تعداد گره‌های اصلی در نظر گرفته می‌شود. در شکل (۴) تعدادی از این درصدها به همراه دقت آن‌ها ذکر شده است. در این شکل دیده می‌شود که بالاترین دقت زمانی به دست می‌آید که ۲۰ درصد از گره‌ها به‌عنوان گره دارای داده جدید در نظر گرفته شوند. در این نسبت جمعیت گره‌های اصلی آن قدر بزرگ است که برآورد مشخصات جمعیت با دقت کافی انجام می‌گیرد. علاوه بر این باقی‌مانده گره‌ها هم به نحو مطلوبی خاصیت جمعیت اصلی را حفظ و منعکس می‌کنند. با افزایش درصد گره‌های جدید، تاثیر مقادیر اندازه‌گیری شده از این گره‌ها بر کل جمعیت هم افزایش می‌یابد و دقت کلی را تحت تاثیر قرار می‌دهد.



شکل (۴). نمودار دقت بر حسب گره های دارای داده جدید

در شکل (۵) تأثیر تعداد گره‌ها بر دقت مشاهده می‌شود. باتوجه به این شکل، افزایش تعداد گره‌ها تا عدد ۲۴ باعث افزایش دقت شده است. این نشان می‌دهد با توزیع کردن داده‌ها در گره‌های بیش‌تر، باز هم کارگزار قادر است با همان دقت یادگیری را انجام دهد. همچنین برای کاهش بار شبکه و زمان محاسبه در ۲۴ گره در مرحله پایانی، طبق قانون گفته شده در بخش‌های قبلی، وزن‌ها سه تا سه تا میانگین گرفته‌شده و به گره‌ها ارسال می‌شوند.



شکل (۵). دقت بر حسب تعداد گره‌ها

در جدول (۲) دقت روش پیشنهادی و دقت چندین روش ارائه شده در مقالات دیگر مقایسه شده‌اند. در این جدول پژوهش‌ها براساس دقتشان رتبه‌بندی شده اند که رتبه آن‌ها در ستون آخر دیده می‌شود. همان‌طور که دیده می‌شود روش پیشنهادی دقت مناسبی را به دست آورده و همچنین از مزایای روش توزیع شده هم استفاده می‌کند. به‌علاوه، به دلیل یادگیری ائتلافی حریم خصوصی گره‌ها حفظ می‌شود. همچنین به دلیل استفاده از توزیع‌شدگی پیچیدگی زمانی روش پیشنهادی نسبت به

روش‌های توزیع نشده از O(N) (تعداد مشتری=N) به O(N/K) (تعداد گره‌ها =K) کاهش یافته است.

جدول (۲). مقایسه دقت روش پیشنهادی با سایر روش‌ها

روش	دقت	رتبه
روش پیشنهادی	۹۶/۴۹	۵
بهاگوات و گوپتا [۴۰]	۹۵/۳	۶
پادمواتی و همکاران [۴۱]	۸۸	۸
یوملمبام و همکاران [۴۲]	۹۸/۳	۴
آتاکاک و همکاران [۴۳]	۹۴/۶	۷
تریپاتی و همکاران [۴۴]	۹۸/۵	۳
شافین و همکاران [۴۵]	۹۹/۴۹	۱
دروس و همکاران [۴۶]	۹۸/۶	۲

۶. بحث

یوملبام و همکاران [۴۲] از روش طبقه‌بندی مبتنی بر شبکه‌های عصبی گرافی برای تشخیص بدافزار استفاده کردند. آن‌ها روش خود را روی مجموعه‌داده‌های CICMalDroid 2020 و Drebin آزمایش کردند که به ترتیب به دقت‌های ۹۸/۳ درصد و ۹۸/۶۸ درصد رسیدند. همچنین برای مقابله با حملات تخاصمی، الگوریتمی مبتنی بر شبکه‌های مولد تخاصمی پیشنهاد دادند.

تریپاتی و همکاران [۴۴] مدلی به نام ADAM را که یک مدل یادگیری ماشین مبتنی بر TensorFlowlite سبک است را توسعه دادند. آن‌ها از یک شبکه عصبی مصنوعی برای تشخیص بدافزار استفاده کردند. مدل آن‌ها با آزمایش بر روی مجموعه‌داده‌ی CICMalDroid 2020 به دقت ۹۸/۵ رسیده است.

شافین و همکاران [۴۵] یک مدل تشخیص مبتنی بر یادگیری ماشین دو لایه بر اساس روش‌های یادگیری جمعی و تعمیم پشته‌ای پیشنهاد داده‌اند تا حملات روزافزون بر روی تلفن‌های هوشمند اندروید را با دقت بیشتری پیش‌بینی و طبقه‌بندی کنند. مجموعه‌داده استفاده‌شده در این پژوهش CICMaldroid 2020 است. این پژوهش با دقت ۹۹,۴۹ درصد، بیشترین دقت را در بین پژوهش‌های پیشین به دست آورده است. در این پژوهش از ویژگی‌های پویا استفاده شده است.

دروس و همکاران از یک طبقه‌بند دودویی برای تشخیص بدافزار بودن یک اپلیکیشن استفاده کرده‌اند [۴۶]. نویسندگان با بکارگیری الگوریتم جنگل تصادفی بر روی مجموعه داده CICMalDroid 2020 توانسته‌اند با دقت ۹۸/۶درصد بدافزارها را از سایر اپلیکیشن‌ها تشخیص دهند. الگوریتم پیاده‌سازی شده در این مقاله یک الگوریتم متمرکز است و از ویژگی‌هایی مانند توزیع‌شدگی و یادگیری افزایشی استفاده نشده است.

این پژوهش‌ها با این‌که دقت بیشتری از روش پیشنهادی به دست آورده‌اند، اما هیچ‌یک از یادگیری ائتلافی و افزایشی استفاده نکرده‌اند؛ بنابراین با هر بار کاهش دقت مدل به دلیل تحول در بدافزارها، این مدل‌ها ناچار هستند یادگیری را از ابتدا انجام دهند. همچنین این پژوهش‌ها صرفاً می‌توانند داده‌ها را از منابعی مانند بازارهای برنامه‌های کاربردی اندروید استخراج کنند که این روش در مقایسه با استفاده از داده‌های تلفن‌های همراه به طور مستقیم، باعث تنوع کمتر داده‌ها می‌شود؛ بنابراین روش پیشنهادی در این پژوهش انعطاف بیشتری را نسبت به سایر پژوهش‌ها دارد.

7. نتیجه گیری

با گسترش تلفن همراه هوشمند، انواع مختلف بدافزارها روزبه‌روز در حال گسترش و توسعه هستند. در این پژوهش از یادگیری ائتلافی - افزایشی برای تشخیص بدافزارها با استفاده از ویژگی‌های ایستا استفاده شده است. مزایایی که از این طریق حاصل می‌شود شامل افزایش دقت، افزایش سرعت نسبت به مدل توزیع نشده، کاهش نیاز به منابع محاسباتی متمرکز (و بنابراین کاهش هزینه) و کاهش بار شبکه نسبت به انجام یادگیری روی تمام داده‌ها از ابتدا است. همچنین مشاهده شد در صورت استفاده از این روش بدون داده جدید، انتخاب ۲۰ درصد گره‌ها برای یادگیری افزایشی یک انتخاب مناسب است. همچنین مشاهده شد افزایش تعداد گره‌ها و میانگین گرفتن از وزن‌ها قبل از ارسال باعث کاهش دقت نشده و افزایش تعداد گره‌ها تا ۲۴ گره حتی دقت را هم‌افزایش می‌دهد؛ بنابراین می‌توان این روش را روشی مقیاس‌پذیر برای تشخیص بدافزارها نام برد. در پژوهش-های آینده می‌توان با استفاده از داده‌های پویا در کنار داده‌های ایستا، انواع بیشتری از بدافزارها را به‌درستی تشخیص داد. همچنین با پیاده‌سازی مدل به‌صورت توزیع‌شده و در محیط واقعی می‌توان درک و ارزیابی بهتری از سرعت یادگیری مدل داشت.

۷. مراجع

[1] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "FED-IIoT: A robust federated malware detection architecture in industrial IoT," IEEE transactions on industrial informatics, vol. 17, no. 12, pp. 8442-8452, 2020.

[2] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A review of android malware detection approaches based on machine learning," IEEE Access, vol. 8, pp. 124579-124607, 2020.

[3] S. Wu, P. Wang, X. Li, and Y. Zhang, "Effective detection of android malware based on the usage of data flow APIs and machine learning," Information and software technology, vol. 75, pp. 17-25, 2016.

[4] R. Jin and B. Wang, "Malware detection for mobile devices using software-defined networking," in 2013 Second GENI research and educational experiment workshop, 2013: IEEE, pp. 81-88.

[5] A. Wang, R. Liang, X. Liu, Y. Zhang, K. Chen, and J. Li, "An inside look at IoT malware," in Industrial IoT Technologies and Applications: Second EAI International Conference, Industrial IoT 2017, Wuhu, China, March 25–26, 2017, Proceedings 2, 2017: Springer, pp. 176-186.

[6] P. Dahiya, "Malware detection in IoT," in Internet of Things: Security and Privacy in Cyberspace: Springer, 2022, pp. 133-164.

[7] S. Qing, "Research progress on Android security," Journal of Software, vol. 27, no. 1, pp. 45-71, 2016.

[8] M. T. Ahvanooey, Q. Li, M. Rabbani, and A. R. Rajput, "A survey on smartphones security: software vulnerabilities, malware, and attacks," arXiv preprint arXiv:2001.09406, 2020.

[9] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," Human-centric Computing and Information Sciences, vol. 8, no. 1, pp. 1-22, 2018.

[10] D. E. García, N. DeCastro-García, and A. L. M. Castañeda, "An effectiveness analysis of transfer learning for the concept drift problem in malware detection," Expert Systems with Applications, vol. 212, p. 118724, 2023.

[11] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket," in Ndss, 2014, vol. 14, pp. 23-26.

[12] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in 2012 IEEE symposium on security and privacy, 2012: IEEE, pp. 95-109.

[13] Y. Mirsky, A. Shabtai, L. Rokach, B. Shapira, and Y. Elovici, "Sherlock vs moriarty: A smartphone dataset for cybersecurity research," in Proceedings of the 2016 ACM workshop on Artificial intelligence and security, 2016, pp. 1-12.

[14] C. I. f. Cybersecurity. "CICMalDroid 2020." https://www.unb.ca/cic/datasets/maldroid-2020.html (accessed 12/23/2023, 2023).

[15] D. Curry. "Android statistics (2023)." https://www.businessofapps.com/data/android-statistics (accessed 12/7/2023, 2023).

[16] P. Faruki et al., "Android security: a survey of issues, malware penetration, and defenses," IEEE communications surveys & tutorials, vol. 17, no. 2, pp. 998-1022, 2014.

[17] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM computing surveys (CSUR), vol. 46, no. 4, pp. 1-37, 2014.

[18] R. Pari, M. Sandhya, and S. Sankar, "A multi-tier stacked ensemble algorithm to reduce the regret of incremental learning for streaming data," IEEE Access, vol. 6, pp. 48726-48739, 2018.

[19] L. U. Memon, N. Z. Bawany, and J. A. Shamsi, "A comparison of machine learning techniques for android malware detection using apache spark," *Journal of Engineering Science and Technology,* vol. 14, no. 3, pp. 1572-1586, 2019.

[20] M. Kumar, "Scalable malware detection system using big data and distributed machine learning approach," *Soft Computing,* vol. 26, no. 8, pp. 3987-4003, 2022.

[21] A. Joshi and S. Kumar, "Stacking-based ensemble model for malware detection in android devices," *International Journal of Information Technology,* vol. 15, no. 6, pp. 2907-2915, 2023.

[22] U. S. Jannat, S. M. Hasnayeen, M. K. B. Shuhan, and M. S. Ferdous, "Analysis and detection of malware in Android applications using machine learning," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2019: IEEE, pp. 1-7.

[23] "Static Analysis of Malware and Benign apps 2017." https://www.kaggle.com/goorax/datasets (accessed 3/9/2024, 2024).

[24] M. Chen, Q. Zhou, K. Wang, and Z. Zeng, "An Android Malware Detection Method Using Deep Learning based on Multi-features," in *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2022: IEEE, pp. 187-190.

[25] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing,* vol. 45, no. 11, pp. 2673-2681, 1997.

[26] H. Javaheri, H. Akbari, and E. Shaghaghi, "Improvement in the Ransomwares Detection Method With New API Calls Features," *Electronic and Cyber Defense,* vol. 8, no. 4, pp. 107-118, 2021.(in Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1399.8.4.9.5%20:DOR

[27] M. Hesabi and M. Deypir, "An Improved Method for Malware Attack Detection in Cloud Computing Using Collective Learning," *Scientific Journal of Electronical & Cyber Defence,* vol. 10, no. 4, 2023. (in Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1401.10.4.4.4

[28] A. Ezzatneshan, T. F. S. Kamel, and R. Ghaemi, "Presentation of a New Solution to Botnet Detection in a Markov Chain-Based Network," 2021.

[29] M. Mosleh and M. Karami, "Presenting a

Malware Detection System by Implementing Hardware Counters Based on the Multi-Layer Perceptron Neural Network (MLP) and the Dragonfly Optimization Algorithm," 2021.

[30] R.-H. Hsu *et al.*, "A privacy-preserving federated learning system for android malware detection based on edge computing," in *2020 15th Asia Joint Conference on Information Security (AsiaJCIS)*, 2020: IEEE, pp. 128-136.

[31] R. Gálvez, V. Moonsamy, and C. Diaz, "Less is More: A privacy-respecting Android malware classifier using federated learning," *arXiv preprint arXiv:2007.08319,* 2020.

[32] B. Lak, V. Yadegari, and A. Matinfar, "dentifying Zero Day Android Daily through Neural Networks," *Electronic and Cyber Defense,* vol. 11, no. 3, pp. 49-55, 2023. (in Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1402.11.3.5.0

[33] M. Ghanavati Nasab, M. Ghazvini, and F. Ghasemian, "Mobile botnets detection using deep learning techniques," *Electronic and Cyber Defense,* vol. 11, no. 2, pp. 31-43, 2023. (in Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1402.11.2.3.6

[34] E. Bastami, H. Soltanizadeh, M. Rahmanimanesh, and P. Keshavarzi, "A Malware Classification Method Using visualization and Word Embedding Features," *Electronic and Cyber Defense,* vol. 11, no. 1, pp. 1-13, 2023. (in Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1402.11.1.1.2

[35] S. Mahdavifar, D. Alhadidi, and A. A. Ghorbani, "Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder," *Journal of network and systems management,* vol. 30, pp. 1-34, 2022.

[36] S. Doraisamy, S. Golzari, N. Mohd, M. N. Sulaiman, and N. I. Udzir, "A Study on Feature Selection and Classification Techniques for Automatic Genre Classification of Traditional Malay Music," in *ISMIR*, 2008: Philadelphia, PA, pp. 331-336.

[37] S. Mahdavifar, A. F. A. Kadir, R. Fatemi, D. Alhadidi, and A. A. Ghorbani, "Dynamic android malware category classification using semi-supervised deep learning," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020: IEEE, pp. 515-522.

[38] S. Mahdavifar, D. Alhadidi, and A. A. Ghorbani, "Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder," *Journal of network and systems management,* vol. 30, no. 1, p. 22, 2022.

[39] K. LIU, et al. A review of android malware detection approaches based on machine learning. IEEE Access, 2020, 8: 124579-124607.

[40] S. Bhagwat and G. P. Gupta, "Android malware detection using hybrid meta-heuristic feature selection and ensemble learning techniques," in *International Conference on Advances in Computing and Data Sciences*, 2022: Springer, pp. 145-156.

[41] G. Padmavathi, D. Shanmugapriya, and A. Roshni, "Performance analysis of unsupervised machine learning methods for mobile malware detection," in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2022: IEEE, pp. 201-206.

[42] R. Yumlembam, B. Issac, S. M. Jacob, and L. Yang, "Iot-based android malware detection using graph neural network with adversarial defense," *IEEE Internet of Things Journal,* 2022.

[43] İ. Atacak, K. Kılıç, and İ. A. Doğru, "Android malware detection using hybrid ANFIS architecture with low computational cost convolutional layers," *PeerJ Computer Science,* vol. 8, p. e1092, 2022.

[44] S. Tripathy, N. Singh, and D. N. Singh, "ADAM: Automatic Detection of Android Malware," in *International Conference on Information Technology and Communications Security*, 2021: Springer, pp. 18-31.

[45] S. S. Shafin, M. M. Ahmed, M. A. Pranto, and A. Chowdhury, "Detection of android malware using tree-based ensemble stacking model," in *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2021: IEEE, pp. 1-6.

[46] A. Droos, A. Al-Mahadeen, T. Al-Harasis, R. Al-Attar, and M. Ababneh, "Android Malware Detection Using Machine Learning," in *2022 13th International Conference on Information and Communication Systems (ICICS)*, 2022: IEEE, pp. 36-41.

*Corresponding Author Email: mysanati@basu.ac.ir

Publisher: Imam Hussein University Authors