Scientific Journal of Electronical & Cyber Defence
Vol. 12, No. 3, Autumn2024, Serial No. 47

DEKF-Driven Few-Shot Class Imbalance Learning to Enhance Cyber Attack Detection

F. Saleh , K. Dadashtabar Ahmadi*, M.A. Keyvanrad,

*Assistant Professor, Imam Hossein University (AS), Tehran, Iran

Received:2024 /05/23, Revised: 2024/09/06, Accepted: 2024/10/07, Published: 2024/10/22

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.3.10.2

ABSTRACT

The escalating use of networks and the internet has led to a surge in cyber threats, making it imperative to develop sophisticated intrusion detection systems (IDS) capable of safeguarding against these malicious intrusions. While machine learning techniques have been extensively employed to enhance IDS, challenges persist, notably in handling imbalanced datasets and rare attack detection such as R2L and U2R due to the small number of their samples in the training dataset. Imbalanced datasets, a common challenge in IDS evaluation, often skew toward majority classes, hindering the detection of minority class attacks. Existing machine learning classifiers, primarily accuracy-driven, struggle to excel at identifying rare attacks, which are often more catastrophic. Moreover, overlapping classes complicate feature selection, further impeding accurate detection. To tackle these challenges, this article proposes a solution rooted in Few-Shot Learning, particularly MAML. Traditional MAML has limitations, including slow convergence and computational demands. To enhance MAML's performance, the article introduces the Node Decoupled Extended Kalman Filter (NDEKF) as an alternative to gradient descent in the inner loop. NDEKF optimizes MAML training, offering faster convergence and improved generalization. The DEKF (Decoupled Extended Kalman Filter) variant simplifies calculations, making it suitable for deep neural networks. The combination of MAML and NDEKF, termed NDEKF-based MAML, is applied to address the imbalanced data problem in IDS. The proposed approach is evaluated on the NSL-KDD dataset, demonstrating its potential to improve rare attack detection in intrusion detection systems. By adopting this approach, we achieved improved convergence speed, enhanced ability to generalize, and higher accuracy compared to the original MAML algorithm when dealing with a sparse and unstable dataset such as NSL-KDD. Particularly, our framework demonstrated significant advancements in accurately detecting rare U2R and R2L attacks. The accuracy rates for R2L and U2R attacks using our proposed framework surpassed those of the original MAML, increasing from 61% to 75% and from 51% to 66%, respectively, even with a reduced number of training epochs.

Keywords: Intrusion detection system, Class Imbalance Learning, Few Shot Learning MAML, Decoupled Extended Kalman Filter

* Corresponding Author Email: dadashtabar@mut.ac.ir

«پدافند الکترونیکی و سایبری»
سال دوازدهم، شماره ۳، فصل پاییز ۱۴۰۳؛ ص ۱۳۷-۱۱۹

آموزش در داده‌های نامتوازن با استفاده از روش آموزش با داده محدود مبتنی بر فیلتر DEKF برای بهبود تشخیص نفوذ حملات سایبری

فتاۀ صالح [۱]، کوروش داداش تبار احمدی [۲*] محمدعلی کیوان‌راد [۳]

۱- دانشجو دکتری ۲-استادیار، ۳- استادیار،دانشگاه صنعتی مالک‌اشتر،تهران، ایران

(دریافت: ۱۴۰۳/۰۳/۰۳، بازنگری: ۱۴۰۳/۰۶/۱۶، پذیرش: ۱۴۰۳/۰۷/۱۶، انتشار: ۱۴۰۳/۰۷/۰۱)
DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.3.10.2

ناشر: دانشگاه جامع امام حسین (ع)

چکیده

درحالی‌که تکنیک‌های یادگیری ماشین به‌منظور بهبود سیستم تشخیص نفوذ حملات سایبری به شکلی گسترده مورداستفاده قرار می‌گیرند، چالش‌های متعدد، به‌ویژه در زمینه مدیریت مجموعه‌داده‌های نامتوازن و تشخیص حملات نادر مانند R2L [۱] و U2R [۲] به دلیل ناکافی بودن تعداد نمونه‌های آنها در مجموعه‌داده‌های آموزشی، به قوت خود باقی هستند. مجموعه‌داده‌های نامتوازن، چالشی رایج و همیشگی در هنگام ارزیابی عملکرد سیستم تشخیص نفوذ بوده و اغلب منجر به انحراف به سمت کلاس اکثریت می‌شوند؛ این عامل به نوبه خود، مانع از شناسایی حملات کلاس‌های اقلیت خواهد شد. طبقه‌بندی‌کننده‌های نیز یادگیری ماشین که عمدتاً مبتنی بر دقت هستند، قادر به شناسایی حملات سایبری نادر نخواهند بود. به‌علاوه، همپوشانی کلاس‌ها انتخاب ویژگی را پیچیده‌تر کرده و مانع تشخیص دقیق نفوذ خواهد شود. ما در این مقاله، برای مقابله با این چالش‌ها، راهکاری ارائه می‌کنیم که منشأ آن در آموزش با داده محدود، به‌ویژه یادگیری متامدل *آگنوستیک* MAML [۳] نهفته است. الگوریتم MAML سنتی محدودیت‌هایی دارد که از آن جمله می‌توان به همگرایی کُند و الزامات محاسباتی اشاره کرد. به‌منظور ارتقای عملکرد MAML، این مقاله فیلتر کالمن گسترش‌یافته جداشده از گره NDEKF [۴] را به‌عنوان جایگزینی برای گرادیان نزولی در حلقۀ داخلی معرفی می‌کند. الگوریتم NDEKF باعث بهینه‌سازی آموزش MAML، تسریع همگرایی و بهبود تعمیم خواهد شد. فیلتر فوق محاسبات را ساده‌تر و آن را برای شبکه‌های عصبی عمیق بهینه‌سازی می‌کند. برای حل معضل داده‌های نامتوازن در سیستم تشخیص نفوذ از ترکیب دو الگوریتم MAML و NDEKF تحت عنوان MAML- NDEKF استفاده شده است. این رویکرد پیشنهادی، بر روی مجموعه‌داده NSL-KDD ارزیابی می‌شود. بعد از اعمال این رویکرد، همگرایی به‌سرعت بهبود می‌یابد، قابلیت تعمیم افزایش پیدا می‌کند و در مقایسه با الگوریتم اصلی MAML، هنگام رویارویی با مجموعه‌داده پراکنده و ناپایداری مانند NSL-KDD دقت بالاتری حاصل می‌گردد. چارچوب پیشنهادی ما به طور خاص، پیشرفت‌های قابل‌توجهی در زمینه تشخیص دقیق حملات R2L و U2R نشان داده است. نرخ دقت تشخیص حملات R2L و U2R در این رویکرد علی‌رغم کاهش تعداد دوره‌های آموزش، بیشتر از MAML اصلی بوده و به ترتیب از ۶۱٪ به ۷۵٪ و از ۵۱٪ به ۶۶٪ افزایش‌یافته است.

کلیدواژه‌ها: سیستم تشخیص نفوذ حملات سایبری، یادگیری نامتوازن کلاس، آموزش با داده محدود MAML، فیلتر کالمن NDEKF

* رایانامه نویسنده مسئول: dadashtabar@mut.ac.ir

[1] Root to Local
[2] User to Root
[3] Model-Agnostic Meta-Learning
[4] Node Decoupled Extended Kalman Filter

۱. مقدمه

طی چند سال گذشته، استفاده سازمان‌ها و افراد از شبکه‌ها و اینترنت به‌منظور برقرار ارتباط و انتقال داده‌ها افزایش چشمگیری داشته است، به‌ویژه از وقتی شیوه‌های کاری با افزایش دورکاری کارمندان دستخوش تغییرات بسیار شد. اما متأسفانه هم‌زمان با این تحول، ریسک حملات سایبری و نفوذهای مخرب نیز افزایش‌یافته است.

گزارش‌های اخیر مرتبط با جرایم سایبری نشان می‌دهد که این نوع حملات در سطح جهان در چند سال گذشته به‌شدت افزایش داشته است. به همین سبب، سازمان‌ها مجبور می‌شوند برای بازیابی اطلاعات، رفع آسیبی که به نام تجاری آنها وارد می‌شود، هزینه‌های عملیاتی، بیمه و غیره میلیون‌ها دلار هزینه کنند.

برای جلوگیری از بروز چنین خطراتی، لازم است سیستم‌های دفاعی پیشرفته و هوشمندی باهدف پیشگیری، شناسایی و پاسخ به این حملات ابداع شوند. در اینجا است که نقش مهم سیستم‌های تشخیص نفوذ (IDS)[1] مطرح می‌شود، چون این سیستم‌ها برای یافتن نشانه‌هایی از فعالیت‌های مخرب موجود، کل ترافیک شبکه را پایش می‌کنند.

چندین دهه است که فن‌ها و روش‌های یادگیری ماشین متعددی برای توسعه این نوع سیستم‌ها مورداستفاده قرار می‌گیرند [۱، ۲]. اما به دلیل وقوع مکرر حملات مخرب به سیستم‌ها و شبکه‌های اطلاعاتی، این موضوع همچنان در دست بررسی و تحقیق است. گرچه اکثر این فن‌ها در زمینه بهبود دقت کلی سیستم‌های تشخیص نفوذ (IDS) موفقیت قابل‌ملاحظه‌ای کسب کرده‌اند، در برخی از حملات نادر یا جدید، میزان دقت آنها چندان رضایت‌بخش نیست و علت آن این است که در اکثر الگوریتم‌های یادگیری ماشین، مجموعه‌داده‌های توزیع متوازن یا هزینه‌های خطای برابر در نظر گرفته می‌شود [۳]. بنابراین، سوگیری[2] مجموعه‌داده‌های نامتوازن حاصله معمولاً به سمت کلاس‌های غالب (اکثریت) و برخلاف کلاس‌های نادر (اقلیت) خواهد بود که به نوبه خود، عملکرد سیستم را تحت‌تأثیر قرار می‌دهد [۴]. این وضعیت "مشکل یادگیری نامتعادل (نامتوازن) یا به‌عبارت‌دیگر، "یادگیری از مجموعه‌داده‌های نامتوازن" نامیده می‌شود.

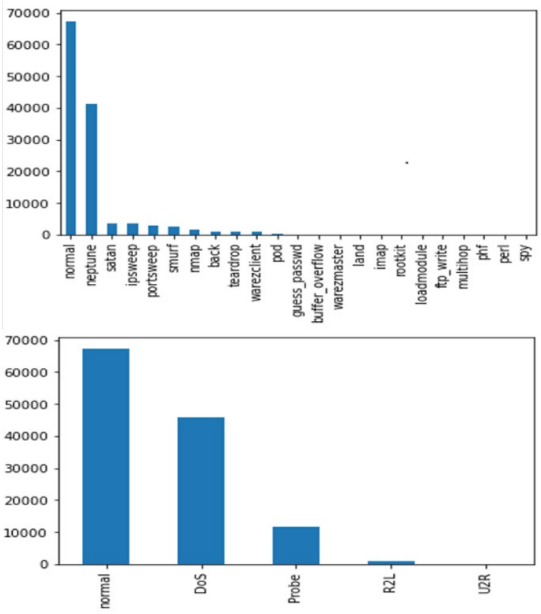
به‌تازگی، یادگیری نامتوازن توجه زیادی را در یادگیری ماشین به خود جلب کرده است، چراکه الگوریتم‌هایی که در حال حاضر پیاده‌سازی می‌شوند، به‌ویژه در خصوص کلاس‌های اقلیت، عملکرد رضایت‌بخشی نشان نمی‌دهند.

به چند دلیل باید چالش جدی تشخیص حملات نادر توسط سیستم‌های تشخیص نفوذ (IDS) را جدی گرفت که مهم‌ترین آنها عبارت‌اند از:

- بخش زیادی از پایگاه‌های داده‌ای که برای ارزیابی عملکرد این سیستم‌ها استفاده می‌شوند، دارای توزیع کلاس چوله (ناموزون) هستند [۵] و تعداد نمونه‌های آموزشی در یک کلاس، به‌شدت از تعداد نمونه‌های آموزشی در کلاس دیگر بیشتر است. به‌عنوان‌مثال، برخی از مهم‌ترین مجموعه‌های داده در زمینه سیستم‌های تشخیص نفوذ (IDS)، مانند KDD-CUP99، NSL KDD، کیوتو یونیورسیتی[3] و دادگان UNB ISCX 2012، نامتوازن هستند. دو دادگان KDD-CUP99 و NSL-KDD شامل دو کلاس حمله کوچک R2L و U2R هستند که تعداد رکوردهای آنها بسیار کمتر از کلاس‌های دیگر است (به شکل (۱) رجوع کنید).
- اکثر طبقه‌بندی‌کننده‌های یادگیری ماشین که در سیستم‌های تشخیص نفوذ (IDS) کاربرد دارند، دقت محور هستند. این طبقه‌بندی‌کننده‌ها در دستیابی به سطوح بالای دقت کلی موفق ظاهر شده‌اند، اما هنوز برای کلاس‌های اقلیت به موفقیت قابل‌قبولی نرسیده‌اند. مثلاً، یک مجموعه‌داده شامل یک کلاس خاص با ۹۸ نمونه و یک کلاس دیگر تنها با ۲ نمونه را در نظر بگیرید. اگر همه داده‌ها برای طبقه‌بندی‌کننده‌ای که از این مجموعه‌داده یاد می‌گیرد، توسط کلاس اول برچسب‌گذاری شوند، این طبقه‌بندی‌کننده همچنان به‌دقت ۹۸٪ دست خواهد یافت [۶].
- ریسک حملات کلاس‌های اقلیت می‌تواند به‌مراتب مخرب‌تر از کلاس‌های اکثریت باشد. هزینهٔ طبقه‌بندی اشتباه این حملات بسیار زیاد است؛ بنابراین، مسئله رسیدن به درصد تشخیص بالا برای این کلاس‌ها برای تضمین قابلیت اطمینان، یکپارچگی و دردسترس‌بودن سیستم‌های شبکه، امری ضروری و چالش‌برانگیز است.
- همپوشانی کلاس‌ها در مرحله انتخاب ویژگی منجر به ایجاد چالش‌های مهمی می‌شود. زیرا ممکن است یک زیرمجموعه ویژگی مشخص نتواند نرخ تشخیص بالایی داشته باشد. تقریباً هدف کلیه تکنیک‌های انتخاب ویژگی اعم از الگوریتم پالایش یا فیلتر و پوشش، یافتن مجموعه ویژگی‌های بهینه برای مجموعه‌داده‌ها است. به‌طورکلی، نتایج کاملاً رضایت‌بخش هستند. هرچند، برای کلاس‌های اقلیت، نرخ تشخیص پایین اعمال شد. دو کلاس R2L و U2R در

[1] Intrusion detection systems
[2] Bais
[3] Kyoto University

دادگان NSL-KDD این نوع مشکل همپوشانی را نشان می‌دهند [۴].



شکل (۱). توزیع کلاس‌های اصلی و کلاس‌های فرعی در دادگان NSL-KDD

تاکنون راهکارهای مختلفی مانند نمونه‌گیری مجدد[1] و یادگیری حساس به هزینه[2] برای حل مسئله یادگیری نامتوازن پیشنهاد شده است [۷]. علی‌رغم این که این راه‌حل‌ها در برخی موارد موفقیت‌آمیز بوده‌اند، مشکل بیش برازش[3] در نمونه‌برداری بیش از حد[4] همچنان به‌چشم می‌خورد، درحالی‌که نمونه‌گیری undersampling به طور بالقوه اطلاعات مفید را حذف می‌کند. علاوه بر این، در این سیستم‌ها الگوریتم‌های یادگیری عمیق[5] بسیاری پیاده‌سازی شده و عملکرد IDS به میزان قابل توجهی بهبود یافته است. با این حال، نتایج حاصل از آنها، از سطوح بالای دقت برای حملات نادر برخوردار نیستند. دلیل این امر آن است که این الگوریتم‌های یادگیری عمیق به مقادیر بسیار زیادی داده برای دستیابی به نسبت‌های تشخیص بزرگ نیاز دارند.

وقتی که صحبت از کاربرد در دنیای واقعی به میان باشد، نمی‌توان همیشه نمونه داده‌های کافی و مورد نیاز برای آموزش یک طبقه‌بندی‌کننده قوی باقابلیت شناسایی هرگونه حمله را به دست آورد. یکی از دلایل این وضعیت آن است که محیط فضای مجازی[6]، محیطی پویا است. دلیل دیگر آن است که همیشه انواع جدیدی از حملات رخ می‌دهند. به‌علاوه اینکه، هیچ نهادی مایل نیست اطلاعات حساس خود را در معرض ریسک این‌گونه حملات قرار دهد. جوامع امنیتی نیز در به‌دست‌آوردن نمونه‌های کافی از حملات در یک دوره زمانی کوتاه‌مدت، با مشکل مواجه هستند [۸]؛ بنابراین، این احتمال وجود دارد که به دلیل مسائل امنیتی یا حساسیت داده‌ها یا حتی هزینهٔ زیاد یا هر دلیل دیگری، هنگام جمع‌آوری تعدادی از نمونه‌های برچسب زده شده، مشکلاتی رخ دهد که به نوبه خود منجر به ایجاد مسئله عدم توازن داده‌ها می‌شود. رفع این محدودیت‌ها در تشخیص نفوذ در دنیای واقعی، مستلزم استفاده از الگوریتم‌های کارآمدتری است که بتوانند مشکل داده‌های محدود برای حملات نادر را به‌منظور آموزش طبقه‌بندی‌کننده‌های نفوذ با قابلیت‌های تعمیم عالی، برطرف نمایند.

به‌غیراز روش‌های مرسومی که در بالا به آن اشاره شد، برای بهبود قابلیت تعمیم IDS در برابر حملات نادر یا حتی حملات جدید و حل چالش توزیع نامتوازن داده‌ها، ما از جدیدترین تکنیک در زمینه یادگیری عمیق یعنی آموزش با داده محدود (FSL)[7] نیز استفاده خواهیم کرد. اگرچه مدت زیادی از عمر تکنیک FSL نمی‌گذرد، اما این روش در مواردی که از مشکل کمبود نمونه‌های برچسب‌گذاری شده مانند تشخیص تصویر و بازشناسی کاراکتر رنج می‌برند، به نتایج قابل‌توجهی دست‌یافته است [۹-۱۳]. علاوه بر این، شبکه‌های عصبی[8] استاندارد قادر به یادگیری وظایف[9] جدید نیستند. طی سال‌های اخیر، محققان با الهام از این که انسان‌ها چگونه می‌توانند مهارت‌های جدید را به‌سرعت یاد بگیرند یا فقط از طریق چند تجربه ساده، با آنها انطباق پیدا کنند، رویکردهای زیادی را برای افزودن قابلیت یادگیری از چند نمونه محدود، به شبکه‌های عصبی عمیق پیشنهاد کرده‌اند. یادگیری متا[10] یا «یادگیری برای یادگیری» یکی از رویکردهایی است که به‌تازگی به‌منظور مقابله با این نوع محدودیت در شبکه‌های عصبی استاندارد مورداستفاده قرار می‌گیرد. در فرایند یادگیری متا، روند یادگیری در دو سطح فرا یادگیری و استراتژی‌های انطباق[11] انجام می‌گردد. در سطح یادگیری متا، مدل موردنظر، مجموعه پارامترهای اولیه مدل را در میان وظایف (taskهای) مختلف یاد می‌گیرد. در طول استراتژی انطباق، این روند، بر مسئله کسب سریع دانش برای یادگیری پارامترهای مخصوص هر وظیفه تمرکز دارد [۱۴]. تاکنون تعداد زیادی از رویکردهای یادگیری متا برای حل مشکلات *آموزش با داده* محدود پیشنهاد شده است. این رویکردها اساساً بسته به نحوه انجام روند بهینه‌سازی فرایند آموزش، با یکدیگر فرق دارند. به‌طورکلی، رویکردهای مورد بحث را می‌توان به سه دسته اصلی

[1] Resampling
[2] Cost sensitive learning
[3] Overfitting
[4] Oversampling
[5] Deep learning
[6] Cyberspace
[7] Few-Shot learning
[8] Neural networks
[9] Task
[10] Meta-learning
[11] Adaptation strategy

طبقه‌بندی کرد: رویکردهای مبتنی بر مدل[1]، مبتنی بر معیار[2] و رویکردهای مبتنی بر بهینه‌سازی[3].

برای مقابله با مشکل داده‌های نامتوازن در IDS و افزایش قابلیت تعمیم مدل در برابر حملات نادر، ما الگوریتم نویدبخش یادگیری متامدل - آگنوستیک (MAML) را ساخته‌ایم [۱۰] که یک رویکرد یادگیری متا مبتنی بر بهینه‌سازی ساده و درعین‌حال مؤثر است. الگوریتم MAML دو مرحله دو در دو را شامل می‌شود. مرحله داخلی برخی از مراحل گرادیان نزولی تصادفی (SGD)[4] را برای هر وظیفه مجزا اجرا و مرحله خارجی، متا پارامتر را در تمام وظیفه‌های نمونه به‌روزرسانی می‌کند. بااین‌حال، در مرحله یادگیری متا از الگوریتم MAML، از الگوریتم یادگیری استاندارد (گرادیان نزولی) در حلقهٔ داخلی استفاده می‌شود. در این مرحله یک یا چند مرحله گرادیان نزولی کامل بر روی تمامی وزن‌ها اجرا می‌گردد. این امر باعث می‌شود این روش با مشکلات عمده‌ای روبرو گردد که مهم‌ترین آنها عبارت‌اند از:

- مقداردهی اولیه پارامترهای مدل مستلزم پس انتشار یا انتشار معکوس[5] از طریق فرایند بهینه‌سازی داخلی با استفاده از چندین مرحله از گرادیان نزولی است. در نتیجه، فرایند یادگیری متا به مشتقات مرتبه بالاتر [۱۵] نیاز دارد که نتیجه آن همگرایی کُند MAML و الزامات محاسباتی و حافظه بسیار است.
- نتیجه استفاده صِرف از وزن‌های نهایی که در حلقهٔ داخلی برای یادگیری حلقهٔ بیرونی محاسبه می‌شوند، عملیات پس انتشار اضافی خواهد بود که می‌تواند منجر به مشکل محوشدگی[6] یا انفجار[7] گرادیان در MAML شود.
- ازآنجایی‌که تمام وزن‌ها از طریق چند مرحله SGD و بر اساس چند نقطه داده (همان داده‌های پشتیبانی) به‌روزرسانی می‌شوند، ممکن است وضعیت بیش برازش اتفاق بیفتد.

برای رفع این محدودیت‌ها در الگوریتم MAML و به‌منظور غلبه بر محدودیت‌های ناشی از تمایز از طریق حلقهٔ درونی در داده‌های کوچک (فرایند تنظیم دقیق یا fine-tuning) که در آن پارامترهای شبکه با استفاده از چند مرحله SGD به‌روزرسانی می‌شوند، چندین راه‌حل پیشنهاد شده است [۱۵-۱۸]. هرچند، این محدودیت‌ها زمانی تأثیرگذارتر می‌شوند که مجموعه‌داده‌های آموزشی، همانند مجموعه‌داده‌های IDS پراکنده و ناپایدار باشند.

تلاش‌های صورت‌گرفته برای بهبود قابلیت تعمیم و سرعت همگرایی MAML نسبت به الگوریتم اصلی MAML، اساساً به کاربرد مراحل SGD در حلقهٔ درونی بستگی دارد که به نوبه خود از مشکلاتی که قبلاً اشاره کردیم رنج می‌برد.

ما برای غلبه بر این محدودیت‌ها در MAML، الگوریتم NDEKF-MAML را توسعه دادیم که در آن، به‌منظور بهینه‌سازی آموزش MAML ، الگوریتم گرادیان نزولی در حلقهٔ درونی با فیلتر کالمن گسترش‌یافته جداشده (NDEKF)[8] جایگزین می‌شود [۱۹]. فیلتر کالمن گسترش‌یافته (EKF)[9] به‌عنوان یک روش برآورد حالت در سیستم‌های غیرخطی، کاربرد فراوانی دارد. رویکرد فیلتر کالمن گسترش‌یافته جداشده برای آموزش شبکهٔ عصبی، یکی از روش‌های مرتبه دوم[10] باتوجه‌به فرضیات ساده‌سازی[11] است [۲۰].

برای افزایش عملکرد همگرایی در مقایسه با الگوریتم پس انتشار، چندین الگوریتم یادگیری شبکهٔ عصبی چندلایه با استفاده از EKF پیشنهاد شده است [۲۱-۲۳]. استفاده از EKF، برآورد حداقل واریانس وزن‌های اتصال[12] را در اختیار ما قرار می‌دهد به‌طوری‌که با سرعتی برابر مرتبه دوم همگرا می‌شود. با این حال، این روش نیز یک اشکال جدی در رابطه با پیچیدگی‌های محاسباتی دارد [۲۴].

اگر همه وزن‌های شبکه در الگوریتم گنجانده شوند، ماتریس‌های خروجی پیچیده شده و مدیریت آن دشوار خواهد بود. به همین دلیل، یک نسخه جدا از گره[13] از این الگوریتم برای ساده‌کردن فرایند محاسبات پیاده‌سازی شده است که عامل مهمی در شبکه‌های عصبی عمیق به شمار می‌رود.

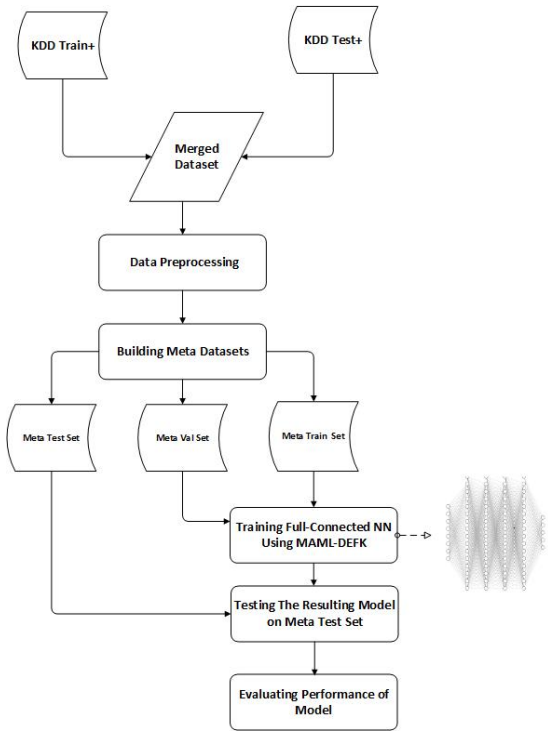
دلیل اصلی این امر آن است که DEKF با تقریب‌زدن داده‌های مرتبه دوم در میان‌وزن‌ها، از EKF به دست می‌آید به‌طوری‌که این داده‌ها فقط به گروه‌های دوبه‌دو ناسازگار تعلق داشته باشند. ممکن است دراین‌بین، همبستگی بین تخمین وزن‌ها نادیده گرفته شود که نتیجه آن، یک ماتریس کوواریانس خطای پراکنده P_k خواهد بود. برای بهینه‌سازی روند آموزش شبکه‌های عصبی، الگوریتم DEKF [۲۴، ۲۵] با موفقیت پیاده‌سازی شده است. در مجموع، DEKF به مراحل آموزشی کمتری نیاز داشته و در مقایسه با الگوریتم‌های گرادیان نزولی ، قابلیت تعمیم بهتری را ارائه می‌دهد.

ایده‌های اصلی در این مقاله بدین شرح هستند. اولین ایده آموزش یک مدل عمیق با داده محدود و با استفاده از روش

[1] Model-based
[2] Metric-based
[3] Optimization-based approaches
[4] Stochastic gradient descent
[5] Backpropagation
[6] Vanishing
[7] Exploding
[8] Decoupled Extended Kalman Filter
[9] Extended Kalman filter
[10] Second-order Methods
[11] Simplifying assumptions
[12] Connection-weights
[13] Node-decoupled version

MAML برای یادگیری حملات سایبری و با چالش دسته‌های نامتوازن از طریق یادگیری تحت نظارت[1] است. در ایده دوم از فیلتر کالمن برای بهینه‌سازی پارامترهای یک شبکهٔ DNN نسبتاً بزرگ استفاده کردیم که از ۴ لایهٔ پنهان هر کدام با اندازهٔ ۳۲ نرون تشکیل شده بود. این کار در هیچ یک از مطالعات قبلی که از فیلتر کالمن برای بهینه‌سازی پارامترهای شبکهٔ عصبی استفاده کرده‌اند، انجام نشده است و در همه کارهای گذشته یک‌لایه و نهایتاً با ۱۰ نرون استفاده شده است. ایده سوم استفاده از فیلتر NDEKF برای بهینه‌سازی پارامترهای شبکهٔ DNN به‌جای گرادیان کاهشی در حلقهٔ داخلی MAML است. با به‌کارگیری چارچوب جدیدمان یعنی NDEKF-MAML توانستیم به بهبود سرعت همگرایی، افزایش قدرت تعمیم و افزایش دقت در مقایسه با الگوریتم اصلی MAML هنگام رسیدگی به مجموعه‌داده‌های نامتوازن مانند NSL-KDD دست یابیم.

روند اجرای کار در این روش در شکل (۲) آمده است. در بخش ۴ توضیح بخش‌های مختلف این شکل به طور کامل شرح داده شده است.



شکل (۲). روندکار MAML-NDEFK پیشنهادی

در نهایت میزان دقت برای این دو حملهٔ اقلیت R2L و U2R با استفاده از چارچوب پیشنهادی ما حتی با کاهش تعداد دوره‌های آموزشی نیز از MAML اصلی پیشی گرفت و به ترتیب از ۶۱٪ به ۷۵٪ و از ۵۱٪ به ۶۶٪ افزایش یافت.

[1] supervised learning

ساختار این مقاله در ادامه بدین شرح است. در بخش ۲ کارهای مرتبط معرفی می‌شوند. در بخش ۳ مدل جدید تشخیص نفوذ MAML-NDEKF ارائه شده و نحوهٔ عملکرد آن به تفصیل بیان می‌شود. در بخش ۴ جزئیات و نتایج تجربی شرح داده می‌شوند. در نهایت در بخش ۵ نیز نتیجه‌گیری و مقایسه با کارهای تحقیقاتی دیگر ارائه می‌شود.

۲. کارهای مرتبط

بر اساس مقدمه فوق و روش پیشنهادی برای حل مشکل شناسایی حملات نادر ناشی از عدم توازن داده‌ها در IDS، از سه مرحله استفاده می‌شود. در مرحله اول روش‌هایی را موردبحث قرار می‌گیرد که برای حل مشکل عدم توازن در مجموعه‌داده‌های IDS مورداستفاده قرار گرفته‌اند. در مرحله دوم، روش‌هایی را که از فیلتر کالمن در آموزش شبکه‌های عصبی استفاده کرده‌اند مورد نقد و بررسی قرار می‌گیرد. در آخرین مرحله نیز از *آموزش با داده* محدود بهره گرفتیم.

۲-۱. روش‌های رفع مشکل عدم‌توزان کلاس در IDS

علی‌رغم این واقعیت که فناوری‌های یادگیری ماشین مبتنی بر IDS قابلیت تشخیص و خودسازگاری قابل‌توجهی را برای جبران تغییرات محیط عملیاتی شبکه فراهم می‌سازد، ولی عملکرد آنها همچنان تحت‌تأثیر توزیع نامتوازن داده‌ها قرار دارد.

بسیاری از مطالعاتی که تکنیک‌های یادگیری ماشین را (خواه سطحی یا عمیق) مورداستفاده قرار داده‌اند (جدول (۱))، نتوانسته‌اند برای حملات نادر و حتی برای حملات سایبری ناشناخته یا مشاهده نشده‌ای که در مجموعه‌داده آموزشی وجود ندارند، به نرخ شناسایی رضایت‌بخشی دست پیدا کنند.

در [۲۶]، محققان یک مدل ترکیبی چند سطحی متشکل از ماشین بردار پشتیبان (SVM)[2] و ماشین یادگیری حداکثری (ELM)[3] را پیشنهاد کرده‌اند. این مدل با استفاده از مجموعه‌داده KDD Cup 1999 ارزیابی شده است. به‌علاوه، در این مدل، یک الگوریتم کی-میانگین[4] اصلاح‌شده برای کاهش اندازه مجموعه‌داده‌های آموزشی و ایجاد توازن در مجموعه‌داده‌های آموزشی برای آموزش SVM و ELM اجرا می‌شود. با این حال، نرخ تشخیص برای کلاس‌های حملات نادر U2R و R2L، به ترتیب معادل ۲۱,۹۳٪ و ۳۱,۳۹٪ بود.

از سوی دیگر، در برخی از مطالعات حملات دارای فراوانی اندک بررسی نشده یا حتی به‌طورکلی کنار گذاشته شده‌اند. در پژوهش [۲۷]، محققان روشی را پیشنهاد کردند که خود رمزگذار

[2] Support vector machine
[3] Extreme learning machine
[4] K-means algorithm

عمیق غیرمتقارن (NDAE)[1] (یادگیری عمیق) را با دقت و سرعت الگوریتم جنگل تصادفی (RF)[2] (یادگیری کم‌عمق) ترکیب می‌کند. آنها از NDAE برای یادگیری روابط پیچیده بین ویژگی‌های مختلف استفاده کردند. علاوه بر این، این روش از قابلیت استخراج ویژگی بالایی برخوردار است که به آن اجازه می‌دهد پارامترهای مدل را با اولویت‌بندی توصیفی‌ترین ویژگی‌ها[3] اصلاح کند. آنها در ادامه از الگوریتم جنگل تصادفی به عنوان طبقه‌بندی‌کننده استفاده کردند. این الگوریتم در مقابل مجموعه‌داده‌های معیار NSL-KDD (همه مجموعه‌داده‌های آموزشی) و KDD Cup '99 (۱۰ درصد از زیر مجموعه برای آموزش) پیاده‌سازی و ارزیابی شده است. آنها مدل خود را بر اساس طبقه‌بندی ۵ کلاسه برای KDD Cup '99 و طبقه‌بندی‌های ۵ کلاسه و ۱۳ کلاسه برای NSL-KDD مورد ارزیابی قرار دادند، درحالی‌که الگوهای حمله سایبری با کمتر از ۲۰ مورد در مجموعه‌داده را برای همه تجربیات حذف کردند. در نهایت، نتایج مدل پیشنهادی و مدل DBN[4] را با یکدیگر مقایسه کردند. آنها مدعی هستند که این مدل یادگیری عمیق، به ترتیب، ضریب دقت ۹۷,۸۵٪ و ۸۵,۴۲٪ را در طبقه‌بندی ۵ کلاسه برای KDD Cup '99 و NSL-KDD و دقت ۸۹,۲۲٪ را برای طبقه‌بندی ۱۳ کلاسه NSL-KDD ارائه می‌دهد.

بر اساس نتایج به‌دست‌آمده، این رویکردها در مقایسه با مدل DBN دقت بالاتری را ارائه کرده و مستلزم صرف زمان کمتری برای آموزش شبکهٔ عصبی هستند.

علاوه بر این، به‌منظور حل مشکل عدم توازن در مجموعه‌داده‌های تشخیص نفوذ، روش‌های کارآمد متعددی با استفاده از تکنیک‌های نمونه‌گیری مجدد پیشنهاد شده است. در پژوهش [۲۸]، مؤلفین باتوجه‌به سطح داده‌ها، یک روش نمونه‌گیری موسوم به RA-SMOTE[5] را برای حل مشکل عدم توازن کلاس پیشنهاد کردند. در این اثر، از سه نوع طبقه‌بندی‌کننده مختلف برای آزمایش اثربخشی الگوریتم استفاده شد که عبارت‌اند از: ماشین‌های بردار پشتیبان (SVM)، شبکهٔ عصبی BP (یا BPNN) و جنگل‌های تصادفی (RF).

نتایج تجربی مربوط به مجموعه‌داده NSL-KDD نشان داد که الگوریتم پیشنهادی می‌تواند به شکلی کارآمد مشکل عدم توازن کلاس را حل کند و نسبت به سایر تکنیک‌های پیشرفته و نوین IDS موجود، نتایج به‌مراتب بهتری را ارائه دهد. در این شیوه، نرخ تشخیص حملات اقلیت U2R و R2L، به ترتیب برابر ۹۸,۶۳٪ و ۹۸,۸۷٪ بود. هرچند، محققان برای ارزیابی عملکرد مدل پیشنهادی خود، به جز نرخ تشخیص، معیار دیگری را در نظر نگرفتند. این معیار به‌تنهایی در مواقع رویارویی با توزیع نامتوازن داده‌ها کافی نیست.

در [۲۹]، نویسندگان سه چارچوب مبتنی بر داده کاوی[6] را پیشنهاد کردند که از الگوریتم جنگل‌های تصادفی (RF) در موارد کاربرد نابجا، ناهنجاری و تشخیص ترکیبی برای بهبود نرخ تشخیص IDS و شناسایی نفوذهای جدید بهره می‌گیرند. برای حل مشکل نفوذهای نامتوازن نیز از نفوذهای اقلیت نمونه‌گیری oversampling و هم از نفوذهای اکثریت نمونه‌گیری undersampling به عنوان یک مرحله پیش‌پردازش استفاده کردند. آنها با استفاده از این چارچوب‌ها در KDD Cup '99 ادعا کردند که مدل پیشنهادی در این پژوهش می‌تواند عملکرد کلی IDSهای فوق را بهبود ببخشد. این درحالی است که در گزارش نهایی آنها هیچ اشاره‌ای به نرخ شناسایی و عملکرد مدل در مورد هر حمله، نشده است. مشکل اصلی مطالعه مورد بحث این است که برخی از نفوذها با درجه تشابه بالا قابل شناسایی نبودند.

در پژوهش [۳۰]، نویسندگان علاوه بر الگوریتم‌های یادگیری متا vote و Random Committee، از شبکهٔ بیزی[7] و الگوریتم درخت تصادفی[8] به عنوان طبقه‌بندی‌کننده اصلی استفاده کردند. مجموعه‌داده KDD Cup '99 نیز برای ارزیابی عملکرد معرفی شده است. اگرچه عملکرد مدل آنها در سطح خوب و قابل قبولی بود، اما میزان دقت آن برای حملات U2R بسیار پایین است.

در [۳۱] مدل تشخیص نفوذ جدیدی به نام ICVAE-DNN ارائه شد. برخلاف استفاده از خودکدگذار[9] متعارف برای یادگیری ویژگی[10]، در این‌جا یک خودکدگذار متغیر شرطی بهبودیافته[11] (ICVAE) برای بیش‌نمونه‌برداری[12] از داده‌های آموزشی به‌منظور رسیدگی به مجموعه‌داده‌های نامتوازن به کار برده می‌شود. ICVAE نمونه‌های حملهٔ جدیدی را برای متوازن‌سازی داده‌ها و افزایش تنوع تولید می‌کند. سپس از این نمونه‌ها برای مقداردهی اولیهٔ وزن‌های یک شبکهٔ عصبی عمیق (DNN) استفاده می‌شود. نتایج تجربی نشان می‌دهند که ICVAE-DNN در مقایسه با پیشرفته‌ترین الگوریتم‌ها و روش‌های بیش‌نمونه‌برداری متعارف در تشخیص حملات اقلیت مانند U2R و R2L دارای عملکرد بهتری است.

[1] Non-Symmetric Deep Auto-Encoder
[2] Random Forest
[3] Descriptive features
[4] Deep Belief Network
[5] Region Adaptive Synthetic Minority Oversampling Technique (RA-SMOTE)
[6] Data mining
[7] Bayesian network
[8] Random Tree
[9] autoencoder
[10] feature learning
[11] Improved Conditional Variational AutoEncoder
[12] oversampling

جدول (۱). خلاصه کارهای مرتبط که به موضوع داده‌های نامتوازن در سیستم‌های تشخیص نفوذ (IDS) می‌پردازد.

سال انتشار مقاله	الگوریتم‌های مورداستفاده برای رسیدگی به مسئلهٔ عدم توازن دسته‌ها	مدل دسته‌بندی‌کنندهٔ مورداستفاده	توضیحات	مجموعه‌داده‌های IDS	نتایج
[۲۶]، ۲۰۱۷	در این کار مطالعاتی، مجموعه‌داده‌های نامتوازن مورد بررسی قرار نگرفتند و صرفاً اندازهٔ مجموعه‌داده‌ها با استفاده از الگوریتم کی - میانگین [۱] اصلاح‌شده کاهش داده شد.	SVM ترکیبی چندسطحی و ELM (ماشین یادگیری افراطی [۲])	نویسندگان مقاله ابتدا الگوریتم کی - میانگین اصلاح‌شده را برای کاهش اندازهٔ مجموعه‌داده‌ها به کار بردند. سپس از یک مدل ترکیبی چندسطحی متشکل از ماشین بردار پشتیبان [۳] (SVM) و ماشین یادگیری افراطی استفاده کردند.	KDD Cup 99 (دسته‌بندی ۵ دسته‌ای)	نرخ تشخیص برای دسته‌های نادر حملات U2R و R2L به ترتیب فقط ۲۱٫۹۳٪ و ۳۱٫۳۹٪ بود.
[۲۷]، ۲۰۱۸	حملات فرکانس پایین مورد بررسی قرار نگرفتند.	استخراج ویژگی با استفاده از روش عمیق NDAE و RF به‌عنوان دسته‌بندی‌کننده	نویسندگان مقاله، خود کدگذار عمیق نامتقارن [۴] انباشته (NSAE) را برای یادگیری روابط پیچیده بین ویژگی‌های مختلف به کار بردند. سپس از جنگل تصادفی [۵] به‌عنوان دسته‌بندی‌کننده استفاده کردند.	KDD Cup 99 (دسته‌بندی ۵ دسته‌ای) و NSL-KDD (دسته‌بندی ۵ دسته‌ای و ۱۳ دسته‌ای)	نتایج حاصل دارای دقت ۹۷٫۸۵٪ و ۸۵٫۴۲٪ در دسته‌بندی ۵ دسته‌ای به ترتیب برای KDD Cup 99 و NSL-KDD و دقت ۸۹٫۲۲٪ برای NSL-KDD ۱۳ دسته‌ای بودند.
[۲۹]، ۲۰۰۸	روش‌های بیش نمونه‌برداری و کم نمونه‌برداری	الگوریتم جنگل‌های تصادفی (RF) در تشخیص سوءاستفاده، ناهنجاری و ترکیبی		KDD Cup 99	در این کار مطالعاتی اشاره نشد که نرخ تشخیص و عملکرد آن برای هر حمله چگونه است.
[۳۱]، ۲۰۱۹	خود کدگذار متغیر شرطی بهبودیافته (ICVAE) به‌عنوان روش استخراج ویژگی و بیش نمونه‌برداری	شبکهٔ عصبی عمیق (DNN)	ICVAE برای یادگیری و بررسی نمایش‌های تُنُک [۶] بالقوه بین ویژگی‌ها و دسته‌های داده‌های شبکه به کار برده می‌شود. کدگشای آموزش‌دیدهٔ ICVAE، نمونه‌های حملهٔ جدیدی را تولید می‌کند. کدگشای آموزش‌دیدهٔ ICVAE برای مقداردهی اولیهٔ وزن لایه‌های پنهان DNN نیز مورداستفاده قرار گرفته است.	NSL-KDD (دسته‌بندی ۵ دسته‌ای) و UNSW-NB15 (دسته‌بندی ۱۰ دسته‌ای)	نویسندگان مقاله ادعا کردند که مقدار DR بالاتری برای U2R و R2L نسبت به مطالعات دیگری که ذکر کرده بودند، به دست آوردند. DR برای U2R و R2L در (KDDTest-21) به ترتیب ۱۱٫۰۰ و ۴۴٫۱۷ بود.

[1] k-means algorithm
[2] Extreme learning machine
[3] support vector machine
[4] Non-Symmetric Deep Auto-Encoder
[5] Random Forest
[6] sparse representations

علی‌رغم پیشرفت‌های قابل‌توجهی که از طریق روش‌های یادگیری عمیق برای ساختن سیستم‌های تشخیص نفوذ کارآمد و مؤثر حاصل شده است، هنگام اعمال بر روی داده‌های نامتوازن، عملکرد ضعیفی از خود نشان داده‌اند. همین امر لزوم بررسی بیشتر رویکردهای جدید برای رسیدگی به این موضوع را برجسته می‌کند. به عقیده ما، به‌جای جمع‌آوری نمونه‌های آموزشی بیشتر که در اکثر کاربردهای دنیای واقعی، عملاً کاری دشوار است، باید به فکر یافتن راهی برای یادگیری از چند نمونه یا حتی بدون نمونه بود؛ بنابراین، فکر کردیم که می‌توانیم با استفاده از الگوریتم آموزش با داده محدود و یادگیری متا باهدف ارتقای سطح عملکرد سیستم تشخیص نفوذ، به هدف نهایی خود در این تحقیق برسیم. اگرچه انجام کار پژوهشی در زمینه یادگیری آموزش با داده محدود، پیشینه زیادی ندارد و عمر این‌گونه تحقیقات به‌سختی به ۱۰ سال می‌رسد، آموزش با داده محدود در زمینه تشخیص کاراکتر[1] [۳۲]، طبقه‌بندی تصویر [۳۳، ۳۴]، تشخیص رویدادهای ویدئویی [۳۵] و طبقه‌بندی متون به نتایج قابل‌توجهی دست‌یافته است. بااین‌حال، تاکنون تنها در یک یا دو مطالعه از آموزش با داده محدود برای تشخیص نفوذ استفاده شده است.

۲-۲. آموزش با داده محدود با استفاده از یادگیری متامدل آگنوستیک (MAML)

الگوریتم MAML [۱۰] یک رویکرد یادگیری متا مبتنی بر بهینه‌سازی است که برای یادگیری پارامترهای اولیه مدل مورداستفاده قرار می‌گیرد به فرایند یادگیری در وظایف جدید از طریق بهینه‌سازی مبتنی بر گرادیان، سرعت می‌بخشد. علاوه بر این، MAML مدل اولیه f_θ به همراه متاپارامتر θ را به عنوان مدل پایه در نظر می‌گیرد. سپس پارامتر θ را می‌آموزد که هنگام استفاده به عنوان پارامترهای اولیه، امکان انطباق سریع‌تر با وظایف جدید را فراهم می‌کند. این الگوریتم از طریق فرایند انطباق، پس‌انتشار می‌یابد و آن را محدود به مدل آموزش با داده محدود می‌سازد، زیرا از نظر محاسباتی پر هزینه بوده و در معرض محوشدگی یا انفجار گرادیان قرار دارد.

برای رفع محدودیت‌های ناشی از تمایز از طریق حلقهٔ داخلی در مجموعه‌داده‌های کوچک (فرایند تنظیم دقیق) در الگوریتم MAML تاکنون نمونه‌های بسیاری پیشنهاد شده است که در آنها پارامترهای شبکه با استفاده از چند مرحله SGD به‌روزرسانی می‌شوند. محققان MAML گونه‌ای از رویکرد پیشنهادی خود را تحت عنوان MAML مرتبه اول (FOMAML)[2] معرفی کردند [۱۰]. در این رویکرد، هنگام تمایز از طریق گرادیان نزولی، مشتقات درجه دوم نادیده گرفته می‌شوند. آنها دریافتند که رویکرد جدید هنگام اعمال بر روی مجموعه‌داده‌های ImageNet، عملکردی مشابه با MAML مرتبه بالاتر دارد. اگرچه پیاده‌سازی روش جدید، ساده‌تر و از نظر محاسباتی مقرون‌به‌صرفه‌تر از MAML اصلی است، این روند ساده‌سازی منجر به ازدست‌دادن اطلاعات گرادیان می‌شود.

یکی دیگر از الگوریتم‌های بهینه‌سازی یادگیری متا، Reptile نام دارد [۱۶] که اساساً الهام‌گرفته از رویکرد FOMAML است. این رویکرد شباهت بسیاری با MAML اصلی دارد، زیرا از طریق اجرای چند مرحله SGD روی تعداد کمی از نمونه‌ها در حلقهٔ درونی، یک مقداردهی اولیه برای پارامترهای شبکهٔ عصبی را می‌آموزد تا به سازگاری سریع‌تر در وظایف جدید و نرخ تعمیم قابل قبولی دست یابد. الگوریتم MAML از طریق فرایند "تنظیم دقیق" به مقادیر وزن اولیه پس‌انتشار می‌یابد، بنابراین محاسبه مشتقات دوم مورد نیاز است. در عوض، رویکرد Reptile گراف محاسباتی را باز نمی‌کند. بلکه بردار وزن مرکزی را در جهت وزن‌های تنظیم شده در طول حلقهٔ بیرونی، به‌روزرسانی و از مشتقات دوم گرادیان‌ها اجتناب می‌کند. علی‌رغم محاسبات مقرون‌به‌صرفه و اجرای ساده Reptile در مقایسه با MAML اصلی بدون در نظر گرفتن مشتقات مرتبه دوم، انجام مراحل SGD در حلقهٔ درونی آن باعث می‌شود کیفیت آن به واسطه مشکلات گرادیان اصلی مانند محوشدگی یا انفجار گرادیان، گیر کردن در بهینه محلی[3] و بیش برازش تهدید شود.

یکی دیگر از نمونه‌های بهبودیافته MAML، الگوریتم MAML++ نام دارد است. محققان در این پژوهش، راه‌حل‌های متعددی را برای مقابله با چالش‌های MAML و بهبود عملکرد تعمیم و سرعت همگرایی آن پیشنهاد کرده‌اند. یکی از راهکارهای پیشنهادی آنها برای غلبه بر مشکل تخریب گرادیان[4] MAML و بهبود پایداری آن، روش بهینه‌سازی ائتلاف چندمرحله‌ای (MSL)[5] است. راهکار MSL به‌جای به‌حداقل‌رساندن تلفات مجموعه هدف پس از محاسبه تمام به‌روزرسانی‌های حلقهٔ درونی، پس از هر بار به‌روزرسانی ائتلاف در مجموعه پشتیبان در حلقهٔ درونی، ائتلاف در مجموعه هدف را به حداقل می‌رساند. یک راه‌حل دیگر که برای بهبود قابلیت تعمیم و سرعت همگرایی MAML پیشنهاد شده است، یادگیری خودکار نرخ یادگیری حلقهٔ درونی است. [۱۷]

۲-۳. فیلتر کالمن برای آموزش شبکه‌های عصبی

برآورد حالت بهینه برای سیستم دینامیکی خطی با استفاده

[1] Character recognition
[2] First-Order MAML
[3] Stuck in local optima
[4] Gradient degradation
[5] Multi-Step Loss Optimization

از فیلترینگ کالمن انجام می‌شود. نسخه‌های توسعه‌یافتهٔ الگوریتم کالمن را می‌توان با خطی‌سازی سیستم حول برآورد فعلی پارامترها برای سیستم‌های دینامیکی غیرخطی به کاربرد. اگرچه الگوریتم کالمن از نظر محاسباتی پیچیده است، اما پارامترها را به‌صورت بازگشتی به‌روز می‌کند که با تمام داده‌های ورودی قبلی سازگار است و معمولاً در چند تکرار همگرا می‌شود. استخراج معادلات کالمن در مراجع زیادی قابل‌دسترسی است [۳۶، ۳۷]. هنگامی که از فیلترینگ کالمن برای آموزش شبکهٔ عصبی استفاده می‌شود، وزن‌های شبکه در واقع پارامترهایی هستند که به‌عنوان حالت‌های سیستم در نظر گرفته می‌شوند و برآورد این حالت‌ها با استفاده از اطلاعات اعمالی به ورودی شبکه انجام می‌شود. شرح مختصری از معادلات کالمنی که با استفاده از آن‌ها پارامترهای شبکهٔ عصبی به‌صورت سازگار محاسبه می‌شوند، در بخش ۲-۳ ارائه می‌شود.

برای رفع اشکالات الگوریتم پس انتشار استاندارد (SBP)[1]، یعنی محوشدگی یا انفجار گرادیان، و بهبود عملکرد همگرایی شبکه‌های عصبی، اعم از نوع پیش‌خور[2] [۳۸، ۳۹] یا بازگشتی[3] [۲۴]، در تکرارها و نمونه داده‌های کمتر نسبت به SBP، چندین الگوریتم یادگیری مشتق‌شده از فیلتر کالمن گسترش‌یافته (EKF) مورد استفاده قرار گرفته است. با این حال، با افزایش پارامترهای شبکهٔ عصبی و اندازه مجموعه‌داده آموزش، تعداد عملیات ماتریسی مورد نیاز برای EKF منجر به افزایش پیچیدگی‌های محاسباتی این رویکردها می‌شود. برای غلبه بر این محدودیت، یک رویکرد تازه با عنوان فیلتر کالمن گسترش‌یافته جداشده (DEFK) در [۲۵] ارائه شده است.

یکی از مزایای اصلی رویکرد DEKF این است که وزن‌های شبکهٔ عصبی را می‌توان به‌گونه‌ای گروه‌بندی (جدا) کرد که عناصر خاصی از $P_{(n)}$ مرتبط با وزن‌های چند گروه، قابل اغماض باشند.

علاوه بر این، اگر گروه‌های وزنی دوبه‌دو ناسازگار باشند، می‌توان $P_{(n)}$ را به شکل یک ماتریس بلوک مورب[4] بازآرایی کرد. اگر با تعداد زیادی از گروه‌ها مواجه باشیم، شاهد کاهش قابل‌توجهی در هزینهٔ محاسباتی و الزامات ذخیره‌سازی خواهیم بود. همچنین، بسته به معیارهایی مانند گره[5]، لایه و یا زیرشبکه[6]، می‌توان گروه‌های وزنی را به شکلی منطقی انتخاب کرد تا زمان کلی آموزش به میزان قابل توجهی کاهش یابد. به طور خلاصه، DEKF علاوه بر قابلیت انتخاب پیچیدگی محاسبات، از کلیه ویژگی‌های GEKF نیز برخوردار است.

تاکنون محققان بسیاری تلاش کرده‌اند از یکی از انواع EKF برای برآورد وزن شبکهٔ عصبی استفاده کنند. به‌عنوان‌مثال، در [۴۰]، برای تشخیص آسیب ساختاری روی یک پل بزرگراه، شبکهٔ عصبی با استفاده از EKF آموزش داده شد. اما به دلیل دشوار بودن محاسبه ماتریس ژاکوبین[7]، از یک لایهٔ پنهان تکین استفاده شد.

برای آموزش شبکه‌های عصبی عمیق با کمک EKF و به‌منظور غلبه بر چالش محاسبه ماتریس ژاکوبین، محققان در [۴۱] الگوریتم جدیدی را برای آموزش شبکهٔ DBN با استفاده از EKF ابداع کردند. این الگوریتم بر اساس محاسبه مشتقات جزئی ماتریس به صورت لایه به لایه، به جای محاسبه مشتق شبکهٔ عصبی در یک معادله واحد استوار است. چارچوب EKF-DBN پیشنهادی در سه نمونه کاربرد مورد آزمایش قرار گرفت. این چارچوب به لحاظ سرعت و دقت محاسبات برای مدل‌سازی تحلیلی پیش‌بینی‌کننده، عملکردی به مراتب بهتر از رویکردهای BP-DBN و RNN از خود نشان داد.

در پژوهش [۴۲]، برای آموزش پارامترهای شبکهٔ LSTM از الگوریتم EKF مستقل استفاده شد. برای کاهش پیچیدگی محاسبه وزن‌ها، هر گره در مدل LSTM به عنوان یک زیرسیستم دینامیک مستقل در نظر گرفته می‌شود که در آن ماتریس کوواریانس[8] حالت، به صورت بلوک مورب تقریب زده می‌شود در حالی که همبستگی بین وزن گره‌های مختلف مانند DEKF با ضمانت عملکرد، نادیده گرفته می‌شود.

بااین‌حال، الگوریتم یادگیری جهانی فیلتر کالمن گسترش‌یافته (GEKF)[9] به دلیل دقت و سرعت همگرایی بالایی که دارد، به شکلی گسترده به‌عنوان یک تکنیک مرتبه دوم، استفاده می‌شود [۱۹]. بااین‌حال، پیچیدگی محاسباتی آن (یعنی $O(m^2)$ که در آن m تعداد وزن‌ها در شبکهٔ عصبی است)، باعث شده است کاربرد این تکنیک برای آموزش شبکه‌های عمیق چالش‌برانگیز باشد، زیرا تعداد پارامترهایی که باید تخمین زده شوند بسیار زیاد است. به‌منظور به حداقل رساندن هزینهٔ محاسباتی GEKF، به جای آن، الگوریتم یادگیری DEKF پیاده‌سازی شده است. الگوریتم DEKF وابستگی متقابل بین وزن‌هایی که به صورت دوبه‌دو به گروه‌های ناسازگار تعلق دارند و در آن وزن‌ها در گره یا لایه گروه‌بندی می‌شوند را نادیده می‌گیرد [۴۳].

[1] Standard backpropagation algorithm
[2] Feedforward Neural Network
[3] Recurrent Neural Network
[4] Block diagonal
[5] Node
[6] Subnetwork
[7] Jacobian matrix
[8] Covariance matrix
[9] Global extended Kalman filter

مطالعات بسیاری، چه به‌صورت تئوری [۴۴] و چه به‌صورت عملی [۴۰، ۴۳، ۴۵] نشان داده‌اند که الگوریتم یادگیری DEKF به لحاظ پایداری و ویژگی‌های همگرایی، عملکردی مشابه با GEKF دارد؛ اما پیچیدگی محاسباتی آن کمتر است. در پژوهش حاضر، ما از فیلتر کالمن گسترش‌یافته از گره (NDEKF) برای محاسبه و به‌روزرسانی وزن‌های سریعی (پارامترهای شبکه) استفاده می‌کنیم که در حلقهٔ درونی (حلقهٔ تنظیم دقیق) الگوریتم MAML با استفاده از چند مرحله SGD محاسبه می‌شوند.

۳. روش پیشنهادی

در این بخش، ابتدا بررسی مختصری بر معادلات الگوریتم دادهٔ محدود MAML خواهیم داشت. سپس، سازوکار مورداستفاده خود برای کاربرد NDEKF به‌عنوان تخمین‌گر وزن‌های سریع در حلقهٔ درونی MAML به‌جای تخمین آنها با استفاده از مراحل SGD را توضیح خواهیم داد.

۳-۱. یادگیری متامدل آگنوستیک (MAML)

یادگیری پارامترهای یک مدل برای یافتن بهترین پارامترهای اولیه، ایده اصلی MAML را تشکیل می‌دهد. به همین جهت، باوجود پارامترهای اولیه خوب، این مدل می‌تواند به‌سرعت وظایف جدید را با استفاده از یک بهینه‌ساز معروف، یعنی SGD، یاد بگیرد. اگر پارامترهای مدل در تکرار N بهینه باشند، تنها یک یا چند تکرار SGD مورد نیاز خواهد بود تا این پارامترها با θ_0 برای طبقه‌بندی وظایف جدید، برازش پیدا کند. به عنوان مثال، برازش پارامترهای θ با تکرار N از طریق به‌حداقل‌رساندن آنتروپی متقاطع[1] در مجموعه پشتیبان انجام می‌شود.

اگر شبکهٔ عصبی f_θ (به عنوان تابعی از متاپارامترهای θ) مدل اصلی باشد، با شروع از مقدار اولیه $\theta = \theta_0$، می‌خواهیم شبکهٔ عصبی را به‌گونه‌ای آموزش دهیم که بتوانیم پس از چند مرحله نسبتاً اندک از به‌روزرسانی گرادیان N با استفاده از مجموعه پشتیبان S_b داده‌ها، در نهایت به وزن‌های سریع θ_N برسیم. این شبکه عملکرد خوبی در مجموعه هدف Q_b دارد، که در آن، b ایندکس یک مجموعه پشتیبانی خاص در گروهی از وظایف مجموعه پشتیبان می‌باشد. ما مجموعه مراحل به‌روزرسانی N را فرایند به‌روزرسانی حلقهٔ درونی می‌نامیم.

پس از تکرار i که در آن $i = 0 \dots N-1$، پارامترهای شبکهٔ عصبی به‌روزرسانی شده این‌گونه تعریف می‌شوند:

$$\theta_{i+1}^b = \theta_i^b - \alpha \nabla_\theta \mathcal{L}_{s_b}(f_{\theta_i^b}) \quad (1)$$

که در آن، α نرخ یادگیری، θ_i^b وزن‌های شبکهٔ مبنا بعد از طی i مرحله به سمت وظیفه b و $\mathcal{L}_{s_b}(f_{\theta_i^b})$ تلفات در مجموعه پشتیبان وظیفه b بعد از (i) مرحله به‌روزرسانی هستند.

اگر فرض کنیم که اندازه گروه وظایف یا task batch برابر B است، آنگاه می‌توانیم یک تابع هدف متا را به این شرح تعریف کنیم:

$$\mathcal{L}_{meta} = \sum_{b=1}^{B} \mathcal{L}_{Q_b}\left(f_{\theta_N^b}(\theta_0)\right) \quad (2)$$

معادله (۱) نشان‌دهنده وابستگی مستقیم θ_N^b به θ_0 است.

معادله (۲) هدف متا را توصیف می‌کند، که در آن، کیفیت θ_0 به عنوان تابعی از کل تلفات ناشی از کاربرد θ_0 به عنوان مقداردهی اولیه برای همه وظایف است.

اکنون ما فرایند به‌روزرسانی حلقهٔ بیرونی را به‌صورت بهینه‌سازی (در اینجا کمینه‌سازی) تابع هدف متا $\mathcal{L}_{meta}$ تعریف می‌کنیم، که به ما اجازه می‌دهد مقدار اولیه بهینه θ_0 را بدست آوریم.

پس از اینکه این مقدار به‌روزرسانی شد، مقدار اولیه بهینه حاصل برای متا پارامترهای θ_0 را می‌توان این‌گونه بیان کرد:

$$\theta_0 = \theta_0 - \beta \nabla_\theta \sum_{b=1}^{B} \mathcal{L}_{Q_b}\left(f_{\theta_N^b}(\theta_0)\right) \quad (3)$$

که در آن، β نرخ یادگیری متا و $\mathcal{L}_{Q_b}$ تلفات در مجموعه هدف برای وظیفه b هستند. انتروپی متقاطع، تابع تلفات[2] (یا هزینه) مشترک برای کلیه عبارت‌های طبقه‌بندی است.

این الگوریتم از دوفاز بهینه‌سازی تشکیل شده است که در آن MAML در حلقهٔ بیرونی، تلفات مورد انتظار هر وظیفه را پس از یک حلقهٔ درونی یا فاز تطبیق، به حداقل می‌رساند و با چند مرحله گرادیان نزولی که در پارامترهای فعلی مدل مقداردهی شده‌اند، محاسبه می‌گردد.

اکنون می‌توانیم با اعمال تغییراتی اندک در برخی از اسامی اصطلاحات، مراحل مختلف این الگوریتم را به‌طورکلی به شکلی که در [۱۵] آمده است، این‌گونه بیان کنیم:

الگوریتم (۱). الگوریتم MAML

Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done do
  3: Sample batch of tasks $T_b = \{S_b, Q_b\} \sim p(T)$
  4: **for all** S_b do
    5: Evaluate $\mathcal{L}_{s_b}(f_{\theta_i^b})$ with respect to K

[1] Cross-entropy

[2] Loss function

examples

6: Compute adapted parameters with N step of gradient descent:

$$\theta_{i+1}^{b} = \theta_{i}^{b} - \alpha \nabla_{\theta} \mathcal{L}_{s_b}(f_{\theta_i^b}) \quad (4)$$

7: **end for**

8: Update $\theta_0 = \theta_0 - \beta \nabla_{\theta} \sum_{b=1}^{B} \mathcal{L}_{Q_b}\left(f_{\theta_N^b}(\theta_{i+1})\right)$

9: **end while**

استفاده از بهینه‌ساز SGD حلقهٔ درونی MAML برای یادگیری وزن‌های سریعی که امکان تطبیق سریع با یک وظیفه جدید را فراهم می‌کند، بسیار ساده است. هرچند، پس انتشار از طریق محاسبات گرادیان متشکل از حلقهٔ درونی و به‌روزرسانی پارامترها در حلقهٔ بیرونی، باعث شده است آموزش MAML به امری دشوار تبدیل شود و با مشکلاتی مثل محوشدگی یا انفجار گرادیان و یا گیرکردن در کمینه‌های محلی دست‌وپنجه نرم کند. به‌علاوه، همین ساختار پیچیده MAML باعث می‌شود انتخاب متا پارامترهای مناسب باتوجه‌به مجموعه‌داده‌ها و سیستم‌های مختلف منجر به افزایش ناپایداری آنها شود.

ایده کاهش وابستگی به عملیات گرادیان در داخل حلقهٔ درونی با جایگزینی بهینه‌ساز SGD با بهینه‌ساز NDEKF از همین‌جا نشئت گرفت؛ بنابراین، ما وزن‌های سریع در حلقهٔ درونی را با استفاده از NDEKF محاسبه می‌کنیم.

البته ایده استفاده از NDEKF برای آموزش شبکه‌های عصبی، اتفاق جدیدی نیست. همان‌طور که در کارهای انجام‌شده پیشین مشاهده کردیم، محاسبه پارامترهای شبکه با استفاده از EKF به لحاظ قابلیت تعمیم و سرعت، عملکرد به‌مراتب بهتری نسبت به SGD دارد، به‌خصوص زمانی که اندازه مجموعه‌داده آموزشی، مانند مورد ما، در حلقهٔ درونی MAML کوچک باشد. هرچند، به دلیل پیچیدگی محاسبه ماتریس ژاکوبین در فیلتر کالمن، به‌ویژه باوجود اندازهٔ بزرگ پارامترها در شبکه‌های عمیق، ماتریس‌های حاصل، به‌شدت غیرقابل‌حل خواهند شد. توصیه می‌شود برای کاهش پیچیدگی محاسبات و هزینه‌های محاسباتی، از یک نسخه "جداشده از گره" از این الگوریتم استفاده شود. ما نیز به همین دلیل الگوریتم NDEKF را به‌عنوان راه‌حلی برای برآورد وزن‌های سریع در حلقهٔ درونی مورداستفاده قرار دادیم.

۳-۲. فرمول‌بندی MAML مبتنی بر NDEKF

بهینه‌سازی وزن‌های شبکهٔ عصبی با استفاده از الگوریتم‌های گرادیان نزولی منجر به کندشدن سرعت همگرایی و ناپایداری آن شده و با مشکل بیش برازش مواجه است، به‌ویژه زمانی که در دو مرحله از رویکردهای یادگیری متا مبتنی بر گرادیان مانند MAML اصلی، روی تعداد کمی از نمونه‌های داده اعمال شود. علاوه بر این، در بهینه‌سازی مبتنی بر گرادیان، مشتقات تابع هزینه باتوجه‌به وزن‌ها، فقط فاصله بین خروجی فعلی و هدف متناظر را در نظر می‌گیرند و اطلاعات تاریخچه در به‌روزرسانی وزن هیچ کاربردی ندارند.

برای اولین‌بار، در پژوهش [۲۵] استفاده از الگوریتم NDEKF برای برآورد وزن شبکهٔ عصبی پیشنهاد شد. NDEFK در مقایسه با الگوریتم EKF به میزان حافظه و زمان محاسبات کمتری نیاز دارد. محققان در آن پژوهش پیشنهاد کردند وزن‌ها با گره جدا شوند، به طوری که فقط وابستگی متقابل وزن‌هایی در نظر گرفته شود که به همان گره تغذیه می‌شوند. در نسخه "جداشده از گره"، از فیلتر کالمن گسترش‌یافته به طور مستقل بر روی هر نورون شبکهٔ عصبی برای تخمین وزن‌های بهینه تغذیه شده روی آن نورون استفاده می‌شود. به این ترتیب، این الگوریتم فقط وابستگی‌های متقابل محلی را مد نظر قرار می‌دهد.

ما در اینجا، به‌اختصار معادلات توصیف‌کننده این فیلتر را بر اساس کار انجام شده در [۴۶] فهرست می‌کنیم. معادلات موجود در پژوهش آنها بر روی شبکه‌ای با عمق سه‌لایه و تعداد کمی نورون در هر لایه اعمال شده است به‌طوری که تعداد آنها از پنج‌لایه تجاوز نمی‌کند. بااین‌وجود، با علم به این واقعیت که امروزه رایانه‌ها قادر به انجام محاسباتی هستند که تا پیش‌ازاین انجام آن ممکن نبود، می‌توان این معادلات را به هر شبکه‌ای با هر عمق و با هر تعداد نورونی تعمیم داد.

الگوریتم EKF، برای تکرار t، بر مبنای نمودی از متغیر حالت به شکل زیر استوار است:

$$\theta_j(t+1) = \theta_j(t) \quad (5)$$

$$y_j(t) = h\left(\theta_j(t).u_j(t)\right) + v_j(t) \quad (6)$$

که در آن، M تعداد نورون‌ها در یک‌لایهٔ خاص است (ما در اینجا نورون‌ها را بر اساس لایه‌ها گروه‌بندی می‌کنیم). به‌علاوه، j یک نورون خاص را تعریف می‌کند. $(1 \le j \le M)$؛ θ_j بردار وزن متناظر با نورون j؛

u_j نشان‌دهنده یک بردار ورودی $n_u \times 1$ برای نورون jام؛

y_i نشان‌دهنده بردار هدف برای نورون خروجی jام در زمان t $(n_y \times 1)$؛

$h(.)$ تابع غیرخطی $\theta(.)$ و $u(.)$ و v_j بردار نویز اندازه‌گیری هستند.

خطای میانگین مربعات (MSE)[1] متناظر با گره j معمولاً با معادله زیر محاسبه می‌شود:

$$\varepsilon = \frac{1}{2}\sum_{j=1}^{n_y}(y_j(t) - \hat{y}_j(t))^2 \quad (7)$$

در مورد وظایف طبقه‌بندی، می‌توان از خطای آنتروپی متقاطع بین خروجی‌های مطلوب و واقعی نیز استفاده کرد:

$$\varepsilon = \frac{1}{2}\sum_{j=1}^{n_y}(y_j(t) - \hat{y}_j(t))^2 \quad (8)$$

معادلات به‌روزرسانی وزن از الگوریتم بازگشتی کالمن که تابع تلفات (معادله ۵ یا ۶) را به حداقل می‌رساند را این‌گونه نوشت:

$$\hat{\theta}_j(t+1) = \theta_j(t) + K_j(t).\varepsilon \quad (9)$$

$$K_j(t) = P_j(t).H_j^T(t).A(t) \quad (10)$$

$$A(t) = \left[R(t) + \sum_{j=1}^{M} H_j(t).P_j(t).H_j^T(t)\right]^{-1} \quad (11)$$

$$P_j(t+1) = P_j(t) - K_j(t).H_j(t).P_j(t) + Q(t) \quad (12)$$

$$P_j(0) = \left(1/\mu_p\right) I \ , \ Q = \mu_q I \ , \ \mathrm{R} = \mu_r I$$

به‌طوری‌که n_j نشان‌دهنده r وزن‌های وارده بر هر نورون j و n_y بعد بردار خروجی،

$x_j(.)$ مقدار تخمینی $\hat{x}_j(.)$؛

$P_j(.)$ ماتریس کوواریانس خطای شرطی تقریبی $n_{x_j} \times n_{x_j}$ در $\theta_j(.)$؛

$R(.)$ ماتریس کوواریانس نویز اندازه‌گیری $(n_y \times n_y)$ موثر بر نرخ همگرایی؛

$K_j(.)$ ماتریس بهره[2] فیلتر کالمن $(n_{x_j} \times n_y)$؛

ε خطای بین خروجی‌های مطلوب و واقعی $n_y \times 1$؛

$Q(.)$ ماتریس کوواریانس مورب نویز فرایند $n_{x_j} \times n_{x_j}$؛

$A(t)$ ماتریسی که یک بار برای هر تکرار محاسبه می‌شود و در همه به‌روزرسانی‌های گره‌های جداشده استفاده می‌شود؛

μ_p. $\mu_q.\mu_r$ هایپرپارامترهای فیلتر که با استفاده از روش آزمون و خطا مقدار دهی اولیه می‌شوند؛

و $H_j(.)$ ماتریس ژاکوبین $n_y \times n_{x_j}$ هستند. این ماتریس همه مشتقات نسبی تابع بردار تعریف‌کننده خروجی y شبکهٔ عصبی متصل به هر وزن وارده بر نورون j را در برمی‌گیرد.

$$H_j(t) = \begin{bmatrix} \frac{\partial y_1}{\partial \theta_{1j}} & \frac{\partial y_1}{\partial \theta_{2j}} & \dots & \frac{\partial y_1}{\partial \theta_{jn_j}} \\ \frac{\partial y_2}{\partial \theta_{1j}} & \frac{\partial y_2}{\partial \theta_{2j}} & \dots & \frac{\partial y_2}{\partial \theta_{jn_j}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial y_{n_y}}{\partial \theta_{1j}} & \frac{\partial y_{n_y}}{\partial \theta_{2j}} & \dots & \frac{\partial y_{n_y}}{\partial \theta_{jn_j}} \end{bmatrix} \quad (13)$$

ما با به‌روزرسانی پارامترها در حلقهٔ درونی با استفاده از چند مرحله DEKF به‌جای گرادیان نزولی، DEKF را با MAML ترکیب می‌کنیم. هرچند، به‌روزرسانی پارامترها در حلقهٔ بیرونی (بهینه‌سازی متا[3]) همانند همان MAML اصلی با استفاده از گرادیان نزولی باقی می‌ماند. برای انجام این کار، ما معادله (۴) در MAML اصلی را با معادلات (۹) تا (۱۲) جایگزین و تغییرات مربوطه را براین‌اساس اعمال می‌کنیم. الگوریتم MAML با استفاده از DEKF برای یک تکرار در حلقهٔ درونی به شرح زیر خواهد بود (به شکل (۳) رجوع کنید):

الگوریتم (۲). الگوریتم پیشنهادی MAML- NDEKF

Require: $p(T)$: distribution over tasks

Require: β: step size hyperparameter

Require: μ_p, μ_q, μ_r: Kalman filer hyperparameters

Require: set $P_i(0) = \left(1/\mu_p\right) I \ , \ Q = \mu_q I \ , \ \mathrm{R} = \mu_r I$

1: randomly initialize θ where $\theta_0 = \theta$

2: **while** not done do

3: Sample batch of tasks $T_b = \{S_b, \ Q_b\} \sim p(T)$

4: **for all** S_b do

5: Compute adapted parameters with N step of DEFK:

6: for depth j of neural network

$$\hat{\theta}_j(t+1) = \theta_j(t) + K_j(t).\varepsilon$$

$$K_j(t) = P_j(t).H_j^T(t).A(t)$$

$$A(t) = \left[R(t) + \sum_{j=1}^{M} H_j(t).P_j(t).H_j^T(t)\right]^{-1}$$

$$P_j(t+1) = P_j(t) - K_j(t).H_j(t).P_j(t) + Q(t)$$

7: **end for**

8: set $\hat{\theta}_i = \left[\hat{\theta}_1^i, \hat{\theta}_2^i, \dots, \hat{\theta}_J^i\right]$ where $0 \le j \le J$, J is a depth of neural network

10: **end for**

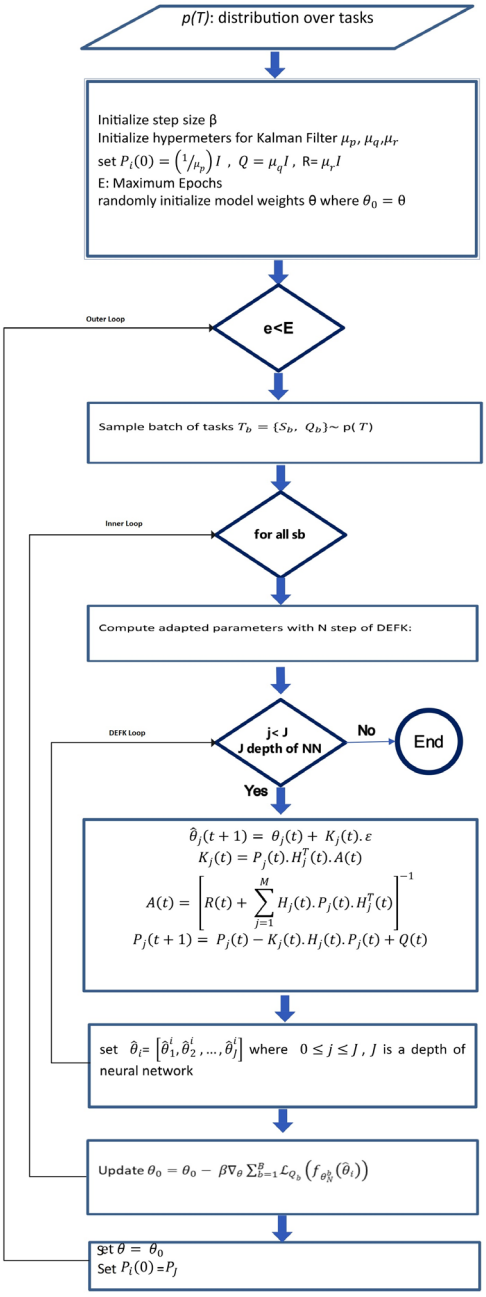
11: Update $\theta_0 = \theta_0 - \beta \nabla_\theta \sum_{b=1}^{B} \mathcal{L}_{Q_b}\left(f_{\theta_N^b}(\hat{\theta}_i)\right)$

12: **end while**

[1] Mean-squared error
[2] Gain matrix
[3] Meta-optimization

بررسی الگوریتم بالا مشخص می‌کند که اجرای الگوریتم DEKF نسبت به کاربرد الگوریتم گرادیان نزولی مستلزم صرف وقت بیشتری است. دلیل این امر آن است که در هر مرحله زمانی، علاوه بر محاسبه مشتقات، محاسبات اضافی بیشتری مانند محاسبه معکوس ماتریس مربع [1] در معادله (۱۱) وجود دارد.

ما به دلیل پیچیدگی، این الگوریتم را در شبکه‌های غیر کانولوشن (غیر پیچشی) [2] اعمال می‌کنیم. ازآنجایی‌که ما این الگوریتم را در دادگان NSL-KDD اعمال می‌کنیم که یک پایگاه‌داده تصویری نیست، عملکرد آن تحت‌تأثیر قرار نخواهد گرفت. برای این منظور از شبکه‌ای با ۴ لایهٔ مخفی با اندازه یکسان ۳۲ استفاده می‌کنیم که هر یک الگوریتم batch normalization و غیرخطی‌های ReLU به‌علاوه، یک لایهٔ خطی و softmax را شامل می‌شوند. ما در مدل خود از آنتروپی متقاطع استفاده کردیم.



شکل (۳). نمودار جریان الگوریتم پیشنهادی

۴. آزمایش‌ها و نتایج

ما در این بخش، عملکرد رویکرد یادگیری عمیق داده محدود پیشنهادی خود را به لحاظ تشخیص نفوذ بهبودیافته مورد ارزیابی قرار می‌دهیم و سپس آن را با رویکرد اصلی MAML مقایسه می‌کنیم. برای این منظور، معیار تشخیص نفوذ پرکاربرد NSL-KDD را پیاده‌سازی خواهیم کرد.

۴-۱. دادگان NSL-KDD

[1] Square matrix
[2] Non-convolutional network

دادگان NSL-KDD [۴۷] یک معیار کارآمد برای تشخیص نفوذ مبتنی بر یادگیری ماشین است. هرچند این دادگان نیز فاقد ایراد نبوده و با مشکل عدم توازن کلاس مواجه است. به‌عنوان‌مثال، در دادگان آموزشی NSL-KDD، تنها ۰٫۰۴٪ از نمونه‌ها به نوع حمله U2R تعلق دارند که باعث می‌شود به‌شدت کمتر از حد نشان داده شود. این وضعیت در مورد انواع حمله R2L و حمله پروب شبکه[1] نیز صادق است، درحالی‌که اکثر سوابق حمله، نشان‌دهنده نوع حمله DDOS هستند.

دسته‌های موجود در مجموعه‌دادهٔ NSL-KDD به پنج دستهٔ اصلی گروه‌بندی می‌شوند:

۱. عادی (Normal)
۲. انکار سرویس[2] (DoS): این حمله با جلوگیری از درخواست‌های قانونی برای دسترسی به منابع شبکه از طریق مصرف پهنای باند یا بارگذاری اضافی بر روی منابع محاسباتی[3] انجام می‌شود.
۳. پویشی (Prob): حملهٔ پویشی، دسته‌ای از حملاتی است که در آن مهاجم قبل از شروع حمله، شبکه را بررسی می‌کند تا اطلاعات سیستم هدف را جمع‌آوری کند.
۴. کاربر به ریشه (U2R): در این حالت، مهاجم با دسترسی به یک حساب کاربری عادی در سیستم اقدام می‌کند و می‌تواند از آسیب‌پذیری‌های سیستم برای دسترسی ریشه-ای[4] به سیستم استفاده کند.
۵. ریشه‌ای به محلی (R2L): مهاجمی که حساب کاربری در یک ماشین راه دور[5] ندارد، بسته‌ای را از طریق شبکه به آن ماشین ارسال می‌کند و از آسیب‌پذیری‌های سیستم برای دسترسی محلی به‌عنوان کاربر آن ماشین استفاده می‌کند.

همین واقعیت، باعث شده است تشخیص انواع حملات کمتر نمایش‌داده‌شده برای طبقه‌بندی‌کننده‌های ML معمولی دشوار شود نوبه خود، منجر به کاهش دقت آن می‌شود. ما برای بهبود دقت در تشخص انواع حملات نادر، به‌غیراز رویکردهای مرسوم رفع مشکل عدم توازن، الگوریتم پرطرف‌دار داده محدود یادگیری متای MAML را به کار گرفتیم و آن را با استفاده از الگوریتم DEKF بهینه‌سازی می‌کنیم. اما پیش از آموزش مدل خود با استفاده از این مجموعه‌داده، مراحل مختلف استاندارد پیش‌پردازش را انجام داده، سپس آن را برای وظایف یادگیری متا تقسیم می‌کنیم.

۴-۲. پردازش NSL-KDD

مجموعه‌داده NSL-KDD از ۴۱ ویژگی، شامل ۳ ویژگی تفکیکی یا رسته‌ای[6] و ۳۸ ویژگی مقادیر پیوسته[7] تشکیل شده است.

انتقال داده: در یادگیری ماشین، نمی‌توان به طور مستقیم از ویژگی‌های رسته‌ای (تفکیکی) استفاده کرد. بلکه برای تبدیل آنها به چند روش رمزگذاری نیاز داریم. روش کدبندی وان هات[8] یکی از متداول‌ترین تکنیک‌های مورداستفاده برای نام‌گذاری ویژگی‌های رسته‌ای است. هر ویژگی از نوع کاراکتر به یک بردار باینری[9] تبدیل می‌شود که در آن هر رسته متناظر با عنوان ۱ یا ۰ برچسب‌گذاری می‌شود. برای مثال، ویژگی protocol_type در مجموع از سه مشخصه tcp، udp، و icmp برخوردار است. با استفاده از روش وان هات، ویژگی tcp به‌صورت (1, 0, 0)، udp به‌صورت (0, 1, 0) و icmp به‌صورت (0, 0, 1) کدبندی می‌شود. به‌طورکلی، سه ویژگی از نوع کاراکتر: *protocol_type*، *service* و *flag*، در مقادیر باینری ۸۴ بعدی نگاشت می‌شوند. در غیر این صورت، مقدار ویژگی *num_outbound_cmds* برابر صفر است و به همین جهت، این ویژگی حذف می‌شود؛ بنابراین، NSL-KDD ۴۱ بُعدی اصلی را می‌توان به یک مجموعه‌داده ۱۲۱ بعدی جدید تبدیل کرد.

نرمال‌سازی داده‌ها[10]: دادگان تبدیل‌شده NSL-KDD از ویژگی‌های ۱۲۱ بعدی با چندین تفاوت در میان آنها برخوردار است. بنابراین، ما از تکنیک نرمال‌سازی داده‌ها برای به‌حداقل‌رساندن این تفاوت‌ها به‌منظور بهبود عملکرد سیستم استفاده می‌کنیم. در این مقاله، روش نرمال‌سازی min–max (معادله (۱۴)) برای کاهش تفاوت‌ها در همه ابعاد و نرمال کردن مقادیر در بازه مشخص [۰،۱] مورد استفاده قرار گرفته است.

$$Normalized_X = \frac{X - X_{min}}{X_{max - X_{min}}} \qquad (14)$$

۴-۳. تقسیم دادگان NSL-KDD برای یادگیری متا

ما مدل‌های داده محدود[11] خود را بر روی توزیع وظیفه FSL (p (T)) ارزیابی می‌کنیم. هر وظیفه FSL در توزیع به‌خودی‌خود، به‌عنوان یک مجموعه‌داده متشکل از دو مجموعه در نظر گرفته می‌شود که مورد اول، مجموعه support و مورد دوم، مجموعه query نامیده می‌شود. مجموعه support برای دربرگرفتن N کلاس و K نمونه از هر یک به کار می‌رود، درحالی‌که مجموعهٔ query، تعمیم مدل به نمونه‌های دیده نشده را حفظ می‌کند؛ بنابراین، مجموعه‌ای متفاوت از نمونه‌های گرفته‌شده از همان N

[1] Probe attack
[2] Denial of Service
[3] computational resources
[4] root access
[5] remote machine

[6] Categorical features
[7] Continuous value features
[8] One-hot-encoding
[9] Binary vector
[10] Data Normalization
[11] Few-Shot

کلاس مجموعه support خواهد بود. همانند یادگیری ماشین با نظارت[1] رایج، ما باید مجموعه‌داده‌های خود را به مجموعه‌های آموزشی، اعتبارسنجی و آزمون تفکیک کنیم. در مورد یادگیری متا، تفاوت در این است که برای جلوگیری از همپوشانی کلاس‌ها و سازماندهی هر مجموعه در وظایف FSL (اپیزودهای مختلف)، به‌جای نمونه‌ها، رده‌ها به مجموعه‌های آموزش متا، اعتبارسنجی متا و آزمون متا تقسیم می‌شوند. برخلاف سایر حوزه‌ها مانند طبقه‌بندی تصویر که در آن معیارهای مختلف FSL مانند Mini-Image Net [۳۳] یا Omniglot [۳۲] وجود دارد، در مورد مجموعه‌داده NSL-KDD، هیچ مجموعه متای کلی برای آموزش با داده محدود نداریم؛ بنابراین، ما مجموعه متا NSL-KDD FSL خود را می‌سازیم. مجموعه‌داده NSL-KDD به نوبه خود از مجموعه‌داده‌های KddTrain+ و KddTest+ تشکیل شده است که محققان برای ساخت مدل‌های یادگیر ماشین و مقایسه عملکرد آنها، به طور گسترده از آنها استفاده می‌کنند. جدول (۲)، توزیع نمونه برای هر کلاس پایه NSL-Kdd در این مجموعه‌داده‌ها را نشان می‌دهد.

علی‌رغم وجود انواع مختلف حملات سایبری در داده‌های آموزشی، توزیع آنها چولگی[2] زیادی دارد. گفتنی است که ترافیک نرمال شامل نیمی از تعداد نمونه‌ها در دو مجموعه‌داده است، درحالی‌که توزیع U2R و R2L بسیار اندک است. بااین‌حال، TestKdd+تعداد بیشتری از نمونه‌های این دو حمله را نسبت به TrainKdd+ شامل می‌شود، به‌علاوه اینکه زیر کلاس‌های جدید (جدول (۳)) این حملات که در TrainKdd+ وجود ندارند را نیز در برمی‌گیرد. همان‌طور که قبلاً اشاره کردیم، در مسائل FSL، مجموعه متا به داده‌های آموزشی زیادی نیاز ندارد و شامل وظایفی است که یک مجموعه support و یک مجموعه query را شامل می‌شود. به همین دلیل، ما برای نمونه‌گیری مجدد وظایف، TrainKdd+و TrainKdd+ را با هم ادغام می‌کنیم و مجموعه متا IDS خود را می‌سازیم. از طرف دیگر، با این ترکیب می‌توانیم به بازنمود خوبی از انواع حملات دست‌یافته و از بروز بیش برازش جلوگیری کنیم.

هر نوع حملهٔ مورد اشاره در جدول (۲)، زیر کلاس‌های مختلفی از تهدیدات را در برمی‌گیرد که در جدول (۳) نشان داده شده است. ما به‌منظور ارزیابی مدل‌های FSL باید داده‌های خود را بر اساس دسته‌ها به مجموعه‌های متای آموزش، اعتبارسنجی و آزمون تقسیم کنیم. علاوه بر ترافیک نرمال، ۳۹ زیر کلاس در داده‌های آموزشی ترکیبی خواهیم داشت که نتیجه آن ۴۰ زیر کلاس کلی است. بنابراین، ما از ۴۰ دسته برای ساخت مجموعه متای خود استفاده خواهیم کرد.

۴-۴. تنظیم[3] مجموعه متا

ما ۴۰ زیرکلاس موجود را به صورت تصادفی به نسبت ۱۰/۱۰/۲۰ به مجموعه‌داده‌های متا آموزش، اعتبارسنجی و آزمون تقسیم کردیم. هرچند، معیار تقسیم ما بر پایه گنجاندن نوع ترافیک نرمال در مجموعه متا آموزشی استوار است. علاوه بر این، ما زیرکلاس‌های مختلفی را از هر چهار حمله در هر مجموعه متا آموزش، اعتبارسنجی و آزمون انتخاب می‌کنیم و در عین حال، شرایطی را فراهم می‌سازیم که بین این مجموعه‌های متا هیچگونه همپوشانی ایجاد نشود. ما در آزمایش خود، به طور تصادفی ۵۰ نمونه از هر زیرکلاس را نمونه‌گیری کردیم.

لازم به ذکر است که ما رویکرد MAML و مدل KF-MAML پیشنهادی خود را در ۱۰ شات ۵ طرفه با اندازه دسته وظایف ۲۵ ارزیابی می‌کنیم.

۴-۵. تنظیم شبکهٔ عصبی و هایپر پارامترها[4]

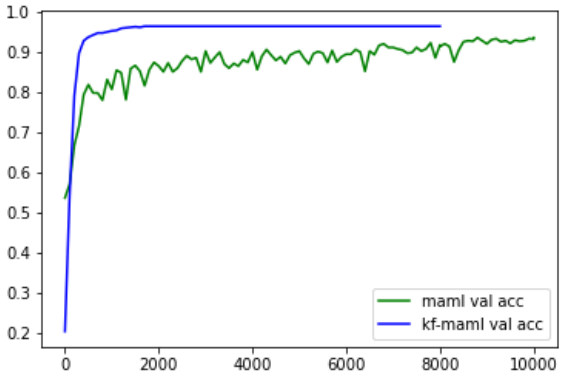
ما برای ارزیابی مدل‌های MAML و KF-MAML، از یک شبکه کاملاً متصل متشکل از ۴ لایهٔ پنهان هر یک با اندازهٔ ۳۲ استفاده می‌کنیم. هر لایهٔ الگوریتم batch normalization (نرمال‌سازی دسته‌ای)، تابع فعال‌سازی ReLU برای غیرخطی‌ها و همچنین یک لایهٔ خطی و یک softmax را شامل می‌شود.

مدل MAML با استفاده از بهینه‌ساز SGD با نرخ یادگیری متا ۰,۰۰۰۲، نرخ یادگیری درونی ۰,۰۱ برای فرایند انطباق و اندازه گام ۴ (یعنی تعداد گام‌های گرادیان) آموزش داده می‌شود.

مدل KF-MAML نیز با استفاده از بهینه‌ساز DEKF متشکل از ۴ مرحله برای تطبیق حلقهٔ درونی، نرخ یادگیری متا ۰,۰۰۰۲ و رابطه $\mu_p = 100$ و $\mu_r = 1$ و $\mu_q = 10e^{-6}$ برای هایپر پارامترهای فیلتر کالمن آموزش داده می‌شود.

همان‌طور که در شکل (۴) نشان داده شده است ، هنگام استفاده از مراحل DEKF در فرایند به‌روزرسانی حلقهٔ درونی MAML به جای گرادیان نزولی در مجموعه‌داده IDS، پس از چند گام زمانی، به نتایج بسیار خوبی دست یافتیم . استفاده از ترکیب DEKF-MAML برای به‌روزرسانی وزن‌ها در حلقهٔ درونی، باعث ارتقای سطح دقت مدل و افزایش سرعت همگرایی آن می‌شود.

[1] Supervised machine learning
[2] Skewed
[3] Setup
[4] Hyper Parameters



شکل (۴). ارزیابی عملکرد مدل اصلی MAML و مدل KF-MAML پیشنهادی ما در مجموعهٔ آزمون متای NSL-Kdd

از سوی دیگر، بر اساس جدول (۴) می‌بینیم که این میزان بهبودی، نه تنها در دقت کلی مدل، بلکه در دقت شناسایی دو حملهٔ نادر R2U و R2L نیز اتفاق افتاده است.

در مطالعهٔ حاضر، ما موفق شدیم فیلتر کالمن را برای بهینه‌سازی یک شبکهٔ عصبی با اندازه نسبتاً بزرگ مورداستفاده قرار دهیم. این کار در مطالعات قبلی مربوط به کاربرد فیلترها در بهینه‌سازی شبکه‌های دارای چندلایهٔ پنهان با تعداد نسبتاً کمی گره انجام نشده بود. علاوه بر این، باتوجه‌به معماری شبکهٔ کاملاً متصل ما با اندازهٔ لایه‌های پنهان ۳۲، هنگام بهینه‌سازی MAML با استفاده از DEKF، سرعت همگرایی و عملکرد مدل به شکل قابل‌ملاحظه‌ای بهبود یافت. از سوی دیگر، رویکرد بهینه‌سازی ما در عمل غیرقابل‌اجرا شده و هنگام اعمال آن در یک شبکهٔ کانولوشن یا زمانی که اندازه هر لایهٔ پنهان ۶۴ باشد، عملکرد آن تنزل پیدا می‌کند. دلیل این امر الزامات محاسباتی زیاد برای محاسبه ماتریس‌ها در فیلتر کالمن است.

برای این‌که کارایی مدل IDS پیشنهادی نشان داده شود، عملکرد ۵ دستهٔ آن با پیشرفته‌ترین IDSها مقایسه شد. جدول (۵) نتایجی را که توسط مدل پیشنهادی به دست آمده است، در مقایسه با سایر مدل‌هایی که بر روی مجموعه‌دادهٔ NSL-KDD آزمایش شده‌اند، از نظر نرخ تشخیص هر دسته و دقت کلی ارائه می‌دهد.

جدول (۲). توزیع نمونه برای هر کلاس پایه NSL-Kdd در مجموعه‌داده‌های آموزش و آزمون

Dataset	Number of Samples					
	Total	Normal	DoS	Probe	U2R	R2L
TrainKdd+	125973	67343 (53%)	45927 (37%)	11656 (9.11%)	52 (0.04%)	955 (0.85%)
TestKdd+	22544	9711 (43%)	7458 (33%)	2421 (11%)	200 (0.9%)	2654 (12.1%)
Total NSL-Kdd	148517	77054 (%52)	53385 (%36)	14077 (9.5%)	252 (0.2%)	3609 (2.3%)

جدول (۳). زیرکلاس‌های هر حمله در مجموعه‌داده NSL-Kdd

Attack Types	DoS	Probe	U2R	L2R
Subclasses	apache2, back, land, neptune, mailbomb, pod, processtable, smarf, treadrop, udpstomp, worm	Ipsweep, mscan, nmap, portsweep, saint,satan	Buffer_overflow, loadmodule, peri, ps, rootkit, sqlattack, xterm	ftp_write, quess_passwd, httptunnel, imap, multihop, named, phf, sendmail, Snmpgetattack, spy, snmpguess, warezmaster, xlock, xsnoop
Total	11	5	7	15

جدول (۴). مقایسهٔ دقت مدل‌های MAML و DEKF-MAML در مجموعهٔ آزمون متای IDS

Approach	Accuracy	DoS	Probe	R2L	U2R	Normal
MAML	0.936	0.98	0.821	0.61	0.51	0.971
DEKF-MAML	0.953	0.951	0.85	0.75	0.662	0.962

جدول (۵): مقایسه بین روش‌های کارهای مرتبط و روش پیشنهادی با استفاده از NSL-KDD برای دسته‌بندی چنددسته‌ای

Approach	Normal	DoS	Probe	U2R	R2L	Accuracy
multi-level hybrid SVM and ELM [26]	**98.13**	**99.54**	**87.22**	21.93	31.39	95.75
S-NDAE [27]	97.73	94.58	72.97	2.70	3.82	85.42
Random forests (RF) algorithm with undersampling and	____	____	____	____	____	94.2

oversampling [29]						
ICVAE-DNN [31]	97.26	85.65	74.97	11.00	44.41	85.97
Original MAML	93.60	98.00	82.10	51.00	61.00	**97.10**
Proposed DEKF-MAML	96.20	95.10	85.00	**66.200**	**75.00**	95.30

روش پیشنهادی از نظر تشخیص نادرترین حملات U2R و R2L دارای افزایش دقت قابل‌توجهی است. همچنین می‌توان مشاهده نمود که باوجود افزایش دقت در این دو حمله، دقت بر روی کل مجموعه‌داده تغییر چندانی نداشته است. شایان‌ذکر است که مدل ما عملکرد قابل تحسینی در سایر دسته‌ها نیز دارد. باتوجه‌به اثر بسیار مخرب این دو حمله به زیرساخت سیستم‌ها، اهمیت این دو بسیار زیاد است، اما متأسفانه روش‌های قبلی نتوانستند به اندازهٔ کافی به آن بپردازند و دقت مناسبی بر روی تشخیص این حملات خطرناک نداشته‌اند. البته روش ما نه فقط باعث افزایش دقت جدی در این دو حمله شده است، بلکه تقریباً در دسته‌های باقیمانده نیز دقت پیشین خود را حفظ کرده است. از طریق انتخاب دقیق هایپرپارامترها[1] و بررسی ساختارهای جایگزین شبکهٔ عصبی عمیق می‌توان به این هدف دست‌یافت. در این کار پژوهشی از یک شبکهٔ عصبی نسبتاً ساده متشکل از چهارلایهٔ پنهان (DNN) استفاده شد.

۵. نتیجه‌گیری

ما در این مقاله، رویکردی نوآورانه در این زمینه تشخیص نفوذ ارائه کردیم که در آن، آموزش داده محدود (FSL) به‌ویژه چارچوب یادگیری متامدل - آگنوستیک (MAML) با فیلتر کالمن گسترش‌یافته جداشده (DEKF) برای بهبود قابلیت‌های سیستم‌های تشخیص نفوذ (IDS) با یکدیگر ادغام شده‌اند.

مدل‌های سنتی یادگیری ماشین که عمدتاً دقت کلی خوبی دارند، اغلب در تشخیص حملات نادر و غالباً فاجعه‌بار، ناکارآمد عمل کرده‌اند. علاوه بر این، وجود کلاس‌های همپوشان، انتخاب ویژگی را پیچیده‌تر کرده و به این چالش می‌افزاید.

ما برای حل این چالش‌ها، راه‌حل جدیدی را معرفی کردیم که از سرعت بالای آموزش با داده محدود با تمرکز ویژه بر مدل MAML بهره می‌برد. ظرفیت MAML در سازگاری سریع با وظایف جدید، حتی باوجود داده‌های برچسب‌گذاری شده محدود، با ماهیت پویای تشخیص نفوذ انطباق کامل دارد.

ما در این مقاله، برای غلبه بر محدودیت‌های سرعت همگرایی و الزامات محاسباتی مدل MAML، فیلتر کالمن گسترش‌یافته جداشده (DEKF) را در عملکرد درونی MAML ادغام کردیم تا جایگزین گرادیان نزولی سنتی شود. این رویکرد ترکیبی که MAML مبتنی بر DEKF نام دارد، با پیشرفت‌های قابل‌توجهی همراه شده و سرعت همگرایی و قابلیت‌های تعمیم مدل را به میزان زیادی بهبود داده است.

ما با انجام ارزیابی دقیق مجموعه‌داده NSL-KDD، اثربخشی رویکرد خود در زمینه بهبود تشخیص حملات نادر در سیستم‌های تشخیص نفوذ را به اثبات رساندیم. نتایج به‌دست‌آمده، بر کارایی این هم‌افزایی میان آموزش با داده محدود و فیلتر کالمن به‌عنوان ابزاری دقیق برای تقویت قابلیت‌های IDS، حتی در مواجهه با تهدیدات سایبری در حال تکامل تأکید دارند.

با استفاده از قابلیت‌های مدل MAML مبتنی بر DEKF، ما گام مهمی در راستای افزایش امنیت شبکه و محافظت از دارایی‌های اطلاعاتی مهم در برابر تهدیدات نوظهور و پیش‌بینی نشده برداشته‌ایم.

باتوجه‌به نتایج اخذ شده و مطالعات تحقیقات این حوزه، روش‌ها و راه‌حل‌های امیدوارکننده‌ای وجود دارد که در آینده می‌توان از آنها بهره جست.

یکی از موضوعات اساسی، بهینه‌سازی رویکرد DEKF-MAML برای مقیاس‌پذیری با تمرکز بر کاهش هزینه‌های محاسباتی و اطمینان از عملکرد درست و الگوریتم شبکه‌های بزرگ‌مقیاس است. توانایی سیستم تشخیص نفوذ ما در مواجهه با گستره روزافزون شبکه‌ها نیز عملکرد درستی خواهد داشت.

چالش‌های مقیاس‌پذیری و کارایی اختلالی در عملکرد سیستم ما نداشته، بلکه آن را در زیرساخت‌های شبکهٔ پیچیدهٔ سازمان‌های مختلف قابل‌دسترس‌تر و کاربردی‌تر می‌کند.

۶. مراجع

[1] Mazloum, H. Bigdeli, "An Optimized Compound Deep Neural Network Integrating With Feature Selection for Intrusion Detection System in Cyber Attacks," Electronic and Cyber Defense, vol. 10, no. 4, pp. 41-51, 2023 (In Persian). https://dorl.net/dor/20.1001.1.23224347.1401.10.4.5.5

[2] M. Hassan Nataj Solhdar, "Investigation of a new ensemble method of intrusion detection system on different data sets," Electronic and Cyber Defense, vol. 10, no. 3, pp. 43-57, 2022 (In

[1] hyperparameters

.3.5.3
[19] S.S. Haykin, and S.S. Haykin, "Kalman filtering and neural networks," vol. 284: Wiley Online Library, 2001. https://doi.org/10.1002/0471221546
[20] B.C. Aissa, and C. Fatima, "Neural Networks Trained with Levenberg-Marquardt-Iterated Extended Kalman Filter for Mobile Robot Trajectory Tracking," Journal of Engineering Science & Technology Review, vol. 10, no. 4, 2017. https://doi.org/10.25103/jestr.104.23
[21] I. Yaesh, and N. Grinfeld, "Training without Gradients--A Filtering Approach," arXiv preprint arXiv:2010.04908, 2020. https://doi.org/10.48550/arXiv.2010.04908
[22] Y. Ollivier, "Online natural gradient as a Kalman filter," Electronic Journal of Statistics, vol. 12, no. 2: pp. 2930-2961, 2018. https://doi.org/10.1214/18-EJS1468
[23] L. Luttmann, and P. Mercorelli, "Comparison of Backpropagation and Kalman Filter-based Training for Neural Networks," 25th International Conference on System Theory, Control and Computing (ICSTCC), 2021. https://doi.org/10.1109/ICSTCC52150.2021.9607274
[24] J.A. Pérez-Ortiz, F.A. Gers, D. Eck, and J. Schmidhuber, "Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets," Neural Networks, vol. 16, no. 2: pp. 241-250, 2003. https://doi.org/10.1016/S0893-6080(02)00219-8
[25] G.V. Puskorius, and L.A. Feldkamp, "Decoupled extended Kalman filter training of feedforward layered networks," IJCNN-91-Seattle International Joint Conference on Neural Networks, 1991. https://doi.org/10.1109/IJCNN.1991.155276
[26] W.L. Al-Yaseen, Z.A. Othman, and M.Z.A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," Expert Systems with Applications, vol. 67: pp. 296-303, 2017. https://doi.org/10.1016/j.eswa.2016.09.041
[27] N. Shone, T.N. Ngoc, V.D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no.1: pp. 41-50, 2018. https://doi.org/10.1109/TETCI.2017.2772792
[28] B. Yan, G. Han, M. Sun, and S. Ye, "A novel region adaptive SMOTE algorithm for intrusion detection on imbalanced problem," 3rd IEEE International Conference on Computer and Communications (ICCC), 2017. https://doi.org/10.1109/CompComm.2017.8322749
[29] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 5: pp. 649-659, 2008. https://doi.org/10.1109/TSMCC.2008.923876
[30] Y. Wang, Y. Shen, and G. Zhang. "Research on intrusion detection model using ensemble learning methods," 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016. https://doi.org/10.1109/ICSESS.2016.7883100
[31] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network," Sensors, vol. 19, no. 11, p.2528, 2019. https://doi.org/10.3390/s19112528
[32] B.M. Lake, R. Salakhutdinov, and J.B. Tenenbaum, "Human-level concept learning through probabilistic program induction," Science, vol. 350, no. 6266: pp. 1332-1338, 2015. https://doi.org/10.1126/science.aab3050

Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1401.10
[3] H. He, and E.A. Garcia, "Learning from imbalanced data," IEEE Transactions on knowledge and data engineering, vol. 21, no. 9. pp. 1263-1284. 2009. https://doi.org/10.1109/TKDE.2008.239
[4] K.-C. Khor, C.-Y. Ting, and S. Phon-Amnuaisuk, "The effectiveness of sampling methods for the imbalanced network intrusion detection data set," Recent Advances on Soft Computing and Data Mining, Springer. pp. 613-622, 2014. https://doi.org/10.1007/978-3-319-07692-8_58
[5] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, vol. 86. pp.147-167. 2019. https://doi.org/10.1016/j.cose.2019.06.005
[6] S. Visa, and A. Ralescu. "Issues in mining imbalanced data sets-a review paper," in Proceedings of the sixteen midwest artificial intelligence and cognitive science conference, vol. 2005, pp. 67-73. sn. 2005.
[7] Y. Liu, T. Gao, and H. Yang, "Selectnet: Learning to sample from the wild for imbalanced data training," Mathematical and Scientific Machine Learning. PMLR. Vol. 107. pp. 193-206. 2020.
[8] H. Liu, and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," Applied Sciences, vol. 9, no. 20: p. 4396, 2019. https://doi.org/10.3390/app9204396
[9] G.S. Dhillon, P.Chaudhari, A. Ravichandran, and S. Soatto, "A baseline for few-shot image classification," in Proceedings of the International Conference on Learning Representations (ICLR), 2020.
[10] C. Finn, P. Abbeel, and S. Levine. "Model-agnostic meta-learning for fast adaptation of deep networks," in Proceedings of 34th International conference on machine learning. vol. 70. pp 1126–1135. 2017. https://dl.acm.org/doi/10.5555/3305381.3305498
[11] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," ICML deep learning workshop. 2015.
[12] H. Tanha, M. Abbasi, "Identify malicious traffic on IoT infrastructure using neural networks and deep learning," Electronic & Cyber Defence, vol. 11, no. 2, 2023 (In Persian). https://dorl.net/dor/20.1001.1.23224347.1402.11.2.1.4
[13] A. A. Tajari Siahmarzkooh, "Intrusion Detection in Computer Networks Using Decision Tree and Feature Reduction," Electronic & Cyber Defence, vol. 9, no. 3, 2021 (In Persian). https://dorl.net/dor/20.1001.1.23224347.1400.9.3.8.9
[14] S. Thrun, and L. Pratt, "Learning to learn: Introduction and overview," in Learning to learn, Springer. pp. 3-17, 1998. https://doi.org/10.1007/978-1-4615-5529-2_1
[15] A. Rajeswaran, C. Finn, S.M. Kakade, and S. Levine, "Meta-learning with implicit gradients," Advances in neural information processing systems, vol. 32, 2019.
[16] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," arXiv preprint arXiv:1803.02999, 2018. https://doi.org/10.48550/arXiv.1803.02999
[17] A. Antoniou, H. Edwards, and A. Storkey, "How to train your MAML," in Proceedings of the International Conference on Learning Representations (ICLR) 2019.
[18] J. Chen, W. Yuan, S. Chen, Z. Hu, and P. Li, "Evo-MAML: Meta-Learning with Evolving Gradient," Electronics, vol. 12, no. 18: p. 3865, 2023. https://doi.org/10.3390/electronics12183865

prediction modeling and efficient data assimilation," Journal of Computing in Civil Engineering, vol. 33, no. 3, 2019. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000835
[42] N.M. Vural, S. Ergüt, and S.S. Kozat, "An efficient and effective second-order training algorithm for lstm-based adaptive learning," IEEE Transactions on Signal Processing, vol. 69 pp. 2541-2554, 2021. https://doi.org/10.1109/TSP.2021.3071566
[43] M.C. Nechyba, and Y. Xu. "Cascade neural networks with node-decoupled extended Kalman filtering," IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', 1997. https://doi.org/10.1109/CIRA.1997.613860
[44] N.M.Vural, F. Ilhan, and S.S. Kozat, "Stability of the Decoupled Extended Kalman Filter Learning Algorithm in LSTM-Based Online Learning," arXiv preprint arXiv:1911.12258, 2019. https://doi.org/10.48550/arXiv.1911.12258
[45] S. Feng, X.Li, S. Zhang, Z. Jian, H. Duan, and Z. Wang, "A review: state estimation based on hybrid models of Kalman filter and neural network," Systems Science & Control Engineering, vol. 11, no. 1, 2023. https://doi.org/10.1080/21642583.2023.2173682
[46] S. Murtuza, and S. Chorian, "Node decoupled extended Kalman filter based learning algorithm for neural networks," 9th IEEE International Symposium on Intelligent Control, 1994. https://doi.org/10.1109/ISIC.1994.367790
[47] M. Tavallaee, E. Bagheri, W. Lu, and A.A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," IEEE symposium on computational intelligence for security and defense applications, 2009. https://doi.org/10.1109/CISDA.2009.5356528
[33] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," Advances in neural information processing systems, 2016. https://dl.acm.org/doi/10.5555/3157382.3157504
[34] J. Snell, K. Swersky, and R. Zemel. "Prototypical networks for few-shot learning," Advances in neural information processing systems, 2017. https://dl.acm.org/doi/10.5555/3294996.3295163
[35] L.Y. Gui, Y.X. Wang, D. Ramanan, and J.M. Moura, "Few-shot human motion prediction via meta-learning," Proceedings of the European Conference on Computer Vision (ECCV). 2018. https://doi.org/10.1007/978-3-030-01237-3_27
[36] B.D. Anderson, and J.B. Moore, "Optimal filtering," Courier Corporation. 2012. https://doi.org/10.1109/TSMC.1982.4308806
[37] A. Gelb (ed.) , "Applied optimal estimation," MIT press. 1974.
[38] Y. Iiguni, H. Sakai, and H. Tokumaru, "A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter," IEEE Transactions on Signal processing, vol. 40, no. 4: pp. 959-966, 1992. https://doi.org/10.1109/78.127966
[39] Y. Shao, F.M. Dietrich, C. Nettelblad, and C. Zhang, "Training algorithm matters for the performance of neural network potential: A case study of Adam and the Kalman filter optimizers," The Journal of Chemical Physics, vol. 155, no. 20, 2021. https://doi.org/10.1063/5.0070931
[40] C. Jin, S. Jang, X. Sun, J. Li, and R. Christenson, "Damage detection of a highway bridge under severe temperature changes using extended Kalman filter trained neural network," Journal of Civil Structural Health Monitoring, vol. 6, no. 3: pp. 545-560, 2016. https://doi.org/10.1007/s13349-016-0173-8
[41] Q. Li, Z.Y. Wu, and A. Rahman, "Evolutionary deep learning with extended Kalman filter for effective