دوازدهمین کنگره ملی سراسری
فناوریهای نوین در حوزه توسعه پایدار ایران
12th National Congress of
the New Technologies in Sustainable Development of Iran
senaconf.ir

پیاده سازی موثر الگوریتمهای یادگیری ماشین بر روی گره‌های لبه در اینترنت اشیاء

پیمان بابائی[۱]، مجید فرشاد[۲]

[۱]گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه آزاد اسلامی واحد تهران غرب، تهران Peyman.Babaei@Iau.ac.ir

[۲]گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه آزاد اسلامی واحد تهران غرب، تهران majidfarshad@gmail.com

چکیده

گره‌های محاسباتی تعبیه‌شده[۱]، به سرعت در حال گسترش در همه جنبه‌های زندگی بشر امروزی هستند و کاربردهای مفیدی در زمینه‌هایی مانند سیستم‌های پوشیدنی برای نظارت بر سلامت، سیستم‌های بی‌سیم برای کنترل نظامی، سیستم‌های شبکه در اینترنت اشیا، لوازم هوشمند برای اتوماسیون خانگی و سیستمهای ترمز ضد قفل در خودروها دارند. روند تحقیقاتی اخیر در فناوریهای محاسباتی، نیاز به پیاده‌سازی روش‌های یادگیری ماشین در گره‌های محاسباتی تعبیه‌شده برای کاربردهای متنوع را نشان می‌دهد. همگرایی الگوریتمهای یادگیری ماشین و سیستم‌های تعبیه‌شده که در آن مدل‌های یادگیری محاسباتی، سیستم‌های تعبیه‌شده با منابع محدود را هدف قرار داده است، فرصت‌های فراوانی را برای تحقیق در فناوری های محاسباتی بر روی گره‌های لبه در اینترنت اشیاء فراهم کرده است. اگرچه تحقیقات در این زمینه در ابتدای راه است، با این حال تحقیقات زیادی شامل بهینه سازی مدل‌های مختلف یادگیری ماشین در محیط‌های با منابع محدود، توسعه معماری‌های سخت افزاری کارآمد شامل واحدهای شتاب دهنده برای تسریع در پیاده‌سازی الگوریتم‌های محاسباتی و معماری‌های سخت‌افزاری تخصصی جدید و نوآورانه برای برآورده نمودن الزامات کارائی در سطوح بالای این مدل‌ها، انجام شده است. لذا، نیاز به بررسی، تجزیه و تحلیل دیدگاه‌های مختلف برای ارائه مفاهیم بنیادی یادگیری ماشین بر روی گره‌های محاسباتی با منابع محدود با در نظر داشتن معماری کامپیوتر هدف، وجود دارد. در مجموع پیاده‌سازی الگوریتم‌های یادگیری ماشین در گره‌های محاسباتی لبه در اینترنت اشیاء نیاز به ارائه راه‌حل‌هایی برای مشکلات پیچیده در این عصر کلان داده دارد. در این مقاله الگوریتمهای یادگیری ماشین اعم مانند درخت تصمیم، k-نزدیک ترین همسایه، رگرسیون لجستیک و جنگل تصادفی مدلسازی شده‌اند. با تحلیل مقایسه‌ای الگوریتمها، درخت تصمیم به عنوان مدلی با عملکرد مناسب با درنظر گرفتن دقت مدل (۹۲.۱۷٪)، میزان حافظه مصرفی و پیچیدگی محاسباتی الگویتم معرفی می شود.

واژه‌های کلیدی

یادگیری ماشین، سیستم‌های تعبیه‌شده، گره های لبه، اینترنت اشیاء.

---

۱ Embedded computing nodes

۱. متن مقاله

فناوری‌های محاسباتی بر روی گره های لبه در اینترنت اشیاء به دلیل پیشرفت‌های جدید در معماری کامپیوتر و به کارگیری الگوریتمهای یادگیری ماشین، در حال تحول هستند. حوزه‌های کاربردی یادگیری ماشین بر روی سیستمهای تعبیه‌شده (بعنوان گره‌های لبه در اینترنت اشیاء)، شامل بینایی ماشین، تشخیص گفتار، مراقبت‌های بهداشتی نوآورانه، رباتیک و سایر موارد بسیار مورد توجه قرار گرفته اند. با این حال، لازم است ساختاری برای پیاده سازی موثر الگوریتم‌های یادگیری ماشین بر روی گره های لبه در اینترنت اشیاء بعنوان بستری مناسب درنظر گرفته شود. الگوریتم‌های یادگیری ماشین عموماً از نظر محاسباتی و حافظه برای محیط‌های دارای محدودیت منابع مانند سیستمهای تعبیه‌شده و سیار نامناسب هستند. برای پیاده‌سازی مؤثر این الگوریتم‌های محاسباتی در فضای محاسباتی تعبیه‌شده و سیار، نیاز به تکنیک‌های بهینه‌سازی نوآورانه در سطوح الگوریتم و سخت‌افزار است. گره های کم مصرف لبه در اینترنت اشیا، مانند میکروکنترلرها، به طور گسترده در کاربردهایی مانند نظارت، پایش سلامت، جستجو و نجات، ردیابی فعالیت، اشکال زدایی ماشین آلات صنعتی، ناوبری داخلی و رباتیک هوایی به کار گرفته شده اند. منابع حافظه و توان محاسباتی تجهیزات لبه در این کاربردها محدود است و نیاز به استقرار سخت‌افزار سبک وزن و خطوط لوله استنتاج دارند. به طور معمول، میکروکنترلرها با استفاده از روش‌های کلاسیک بر روی داده‌های ساختار یافته دریافتی از سنسورها با ابعاد پایین (مانند دما و رطوبت) کار می‌کنند و استنتاج‌های ساده‌ای را در لبه انجام می‌دهند. اخیراً، تلاش‌های قابل توجهی برای پیاده سازی الگوریتمهای یادگیری ماشین در لبه در حال انجام است. با این حال، مدل‌های یادگیری ماشین که مستقیماً برای تجهیزات لبه طراحی شده‌اند، مانند تلفن‌های همراه و یا رایانه‌های تک برد، برای میکروکنترلرها مناسب نیستند. یک میکروکنترلر معمولی دارای ۱۲۸ کیلوبایت حافظه رم و ۱ مگابایت فلش است، در حالی که یک تلفن همراه می تواند ۴ گیگابایت حافظه رم و ۶۴ گیگابایت حافظه ذخیره سازی داشته باشد. محدودیت منابع گره‌های لبه اینترنت اشیاء، طراحی یک گردش کار سیستماتیک و ابزارهایی را برای استقرار خطوط لوله یادگیری ماشین نیاز دارد. در این مقاله الزامات، چالش‌ها و فرصت‌هایی را در توسعه مدل‌های یادگیری ماشین برای پردازش اطلاعات حسگرها بر روی میکروکنترلرها ارائه می کنیم. برای درک اینکه چگونه تجهیزات محاسباتی تعبیه‌شده کم مصرف ممکن است به پر کردن تقاضای فوق‌الذکر کمک کنند، الگوریتمهایی را که نیازمندی‌های اینترنت اشیاء را با اجرای الگوریتمهای یادگیری ماشین با مصرف انرژی کارآمد برای تجهیزات کم مصرف برآورده می‌کنند، بررسی می‌کنیم.

۲. یادگیری ماشین و محاسبات لبه

الگوی محاسبات لبه در اینترنت اشیا، با استفاده از تجهیزات هوشمند، فرآیند محاسباتی را به منابع داده مانند سنسورهای محیطی و دوربین‌ها نزدیک‌تر می‌کند. در چند سال اخیر، تحقیقاتی در زمینه تجزیه و تحلیل، تصمیم گیری و کنترل با گره های محاسباتی لبه که الگوریتم های کامل را اجرا می کنند، انجام شده است و تجهیزات لبه هوشمند کم مصرف با منابع حافظه و توان محاسباتی محدود که الگوریتم‌های یادگیری ماشین را برای شناسایی اشیاء و یا استخراج ویژگی‌ها اجرا می‌کنند محقق شده اند. پارادایم محاسبات لبه یک الگوی نوظهور برای سیستم‌های اینترنت اشیا است که در آن محاسبات در انواع تجهیزات لبه توزیع می‌شوند تا قابلیت محاسباتی را به منابع داده، مانند حسگرهای محیطی و دوربین‌ها نزدیک‌تر کنند. می توان به مزایای محاسبات لبه نسبت به الگوی محاسبات متمرکز سنتی که در سیستم های ابری یافت می شود اشاره کرد:

- کاهش پهنای باند ارتباطی و هزینه های انرژی، در نتیجه کاهش انتقال داده ها به سرورهای ابری متمرکز.
- نزدیکی فیزیکی داده‌ها و تجهیزات پردازشی، لذا پردازش داده‌ها در زمان واقعی تسهیل می‌شود.
- پردازش در تجهیزات لبه، حریم خصوصی داده‌های حساس را تضمین می‌کند و از بارگذاری آنها به مکانهای دور جلوگیری می‌کند.
- در سیستمهای توزیع شده، خرابی برخی از گره ها را می توان به راحتی با حداقل تاثیر بر کل سیستم برطرف نمود و تجهیزات جدید را به صورت ماژولار برای افزایش توان محاسباتی اضافه کرد.

برای گره های محاسباتی لبه، تجزیه و تحلیل داده ها، به ویژه پیاده سازی الگوریتمهای یادگیری ماشین، یک چالش بزرگ است. امروزه کاربردهایی که می توانند از تکنیک های یادگیری ماشین در لبه استفاده کنند، بسیار هستند. ولی آنها با مشکل استنتاج سروکار دارند. بعنوان چند نمونه کاربرد می توان به تجزیه و تحلیل یک ویدیوی بلادرنگ در سیستم های نظارت تصویری کنترل ترافیک و یا وسایل

نقلیه خودران، استخراج ویژگی از تصاویر جهت شناسایی نواحی و اشیاء، شناسایی کاراکترهای دست نویس، نظارت بر مراقبت های بهداشتی و به حداقل رساندن مصرف انرژی با استفاده از منابع تجدیدپذیر در خانه و شهر هوشمند اشاره نمود.

گره های محاسباتی تعبیه‌شده، به سرعت در حال گسترش در همه جنبه‌های زندگی بشر امروزی هستند و کاربردهای مفیدی در زمینه‌هایی مانند سیستم‌های پوشیدنی برای نظارت بر سلامت، سیستم‌های بی‌سیم برای نظارت نظامی، سیستم‌های شبکه در اینترنت اشیا، لوازم هوشمند برای اتوماسیون خانگی و سیستم های ترمز ضد قفل در خودروها دارند[۱،۲]. روند تحقیقاتی اخیر در فناوری های محاسباتی، نیاز به پیاده سازی روش های یادگیری ماشین در گره های محاسباتی تعبیه‌شده برای کاربردهای متنوع را نشان می دهد. پیشرفت‌های اخیر، کاربرد مدل‌های یادگیری ماشین را در زمینه‌های کاربردی مانند بینایی کامپیوتر[۳-۷]، تشخیص گفتار[۹،۸]، ترجمه و پردازش زبان طبیعی[۱۱،۱۰]، رباتیک و مراقبت‌های بهداشتی [۱۲] نشان می‌دهد. روش های یادگیری ماشین مانند بردار ماشین پشتیبان (SVM) برای طبقه بندی ویژگی ها[۱۳]، شبکه های عصبی کانولوشنی (CNN) برای تشخیص نفوذ[۱۴]، و نیز تکنیک های یادگیری عمیق، به منابع محاسباتی و حافظه بالایی برای آموزش و استنتاج موثر نیاز دارند[۱۵-۱۸]. واحدهای پردازشی همه منظوره، حتی با اصلاح معماری آنها در طول سالها، از جمله خط لوله، سلسله مراتب حافظه پنهان عمیق، پیشرفت های چند هسته ای و غیره، نمی توانند نیاز محاسباتی بالای مدل های یادگیری را برآورده کنند. با این حال، واحدهای پردازش گرافیکی (GPU)، به دلیل عملیات ممیز شناور و پردازش موازی در سطح نخ، برای آموزش مدل‌های یادگیری عمیق مناسب‌تر هستند[۱۹]. تحقیقات گسترده‌ای برای توسعه واحدهای شتاب دهنده سخت‌افزاری مناسب با استفاده از ASICs، GPUs، FPGAs و TPUs [۲۰-۲۶]. برای ایجاد سیستم‌های ناهمگن و گاها" توزیع‌شده برای پاسخگویی به تقاضای توان محاسباتی بالای مدل‌های یادگیری در حال انجام است. در هر دو سطح الگوریتم و سخت‌افزار، تکنیک‌های بهینه‌سازی برای پیاده سازی الگوریتمهای یادگیری ماشین کلاسیک و الگوریتم‌های یادگیری عمیق در حال بررسی هستند تا امکان اجرای کارآمد مدل‌های یادگیری ماشین در تجهیزات سیار و سیستمهای تعبیه‌شده فراهم شود [۲۷-۲۹].

سرجیو و همکاران[۲۴] یک بررسی جامع در خصوص پیاده سازی الگوریتم های یادگیری ماشین در سیستم‌های تعبیه‌شده و سیار انجام داده‌اند و مفاهیم و تکنیک‌های یادگیری ماشین را برای بهینه‌سازی و ارائه در حوزه‌های کاربردی مختلف بررسی کرده‌اند. قابل توجه است که چندین تحقیق برای دستیابی به این امر انجام شده است. برای مثال مراجع [۳۱،۳۰] مفاهیم، مدل‌ها و بهینه‌سازی‌های یادگیری عمیق را بررسی می‌کنند. در این تحقیق ها، توجه کمی به طراحی معماری سخت‌افزار شده است، که یک نگرانی کلیدی در توسعه سیستم‌های یادگیری ماشین کارآمد است. پوجا [۳۲] روند اخیر در طراحی معماری سخت افزار را برای برنامه های کاربردی یادگیری ماشین، با استفاده از واحد پردازش تنسور به عنوان مطالعه موردی بررسی می کند. با این حال، این تحقیق معماری‌های مختلف شبکه های عصبی عمیق را مورد بررسی قرار نداده است و فقط برخی از تکنیک‌های بهینه‌سازی یادگیری عمیق را بررسی کرده است. جیاسی و ژوکان [۳۳]، مفاهیم یادگیری عمیق را بررسی کردند، استنتاج در گره های محاسباتی را محدود کردنده اند اما معماری‌های تعبیه‌شده مختلف را برای اطلاع از اینکه کدام معماری یا بهینه‌سازی برای مدل‌های مختلف شبکه های عصبی عمیق مناسب است، مقایسه نکرده اند و فقط یک مدل یادگیری عمیق را بررسی کرده و دیگر مدل های یادگیری عمیق (CNN، RNN) را که مورد توجه قرار گرفته اند، تحقیق نکرده اند.

۳. تکنیک‌های یادگیری ماشین تعبیه‌شده

یادگیری ماشین شاخه‌ای از هوش مصنوعی است که تکنیک‌هایی را توصیف می‌کند که از طریق آن سیستم‌ها از داده‌های موجود یاد می‌گیرند و هوشمندانه تصمیم می‌گیرند. تکنیک های یادگیری ماشین را می توان در سه گروه عمده طبقه بندی کرد که عبارتند از یادگیری نظارت شده، یادگیری بدون نظارت، و یادگیری تقویتی. در یادگیری نظارت شده، داده های برچسب گذاری شده را می توان آموخت در حالی که در یادگیری بدون نظارت، الگوهای پنهان را می توان از داده های بدون برچسب کشف کرد و در یادگیری تقویتی، یک سیستم ممکن است از محیط اطراف خود از طریق روش آزمون و خطا یاد بگیرد[۳۴-۳۶]. فرآیند یادگیری به عنوان مرحله آموزش مدل نامیده می شود و اغلب با استفاده از معماری های کامپیوتری با منابع محاسباتی بالا مانند پردازنده های گرافیکی متعدد انجام می‌شود. پس از یادگیری، مدل آموزش دیده برای تصمیم گیری هوشمندانه در مورد داده های جدید استفاده می شود. از این فرآیند به

عنوان مرحله استنتاج پیاده سازی یاد می شود. این استنتاج اغلب در دستگاه های کاربر با منابع محاسباتی محدود مانند تجهیزات اینترنت اشیا و دستگاه های تلفن همراه انجام می شود.

در یادگیری نظارت شده، مسئله طبقه بندی و رگرسیون اغلب با الگوریتم هایی مانند ماشین بردار پشتیبان (SVM)، شبکه های عصبی مصنوعی (ANN) و رگرسیون خطی به دست می آید. با استفاده از الگوریتم هایی مانند k-means می توان از یادگیری بدون نظارت برای مسائل خوشه بندی و پیش بینی استفاده کرد. اکثر الگوریتم‌های یادگیری ماشین که بر روی تجهیزات لبه پیاده‌سازی می‌شوند، در حال حاضر از استنتاج (فرآیند حل مستقیم مسائل یادگیری ماشین با شبکه‌های عصبی از قبل آموزش‌دیده شده) به جای آموزش (فرآیند به حداقل رساندن خطا به عنوان تابعی از پارامترهای شبکه عصبی، با توجه به مسئله یادگیری ماشین داده شده) استفاده می‌کنند. یکی از دلایل ذکر شده پهنای باند و هزینه‌های تأخیر بالا است که در تبادل داده ها و بروزرسانی‌های شبکه در تجهیزات متعدد لبه وجود دارد (آموزش مدل متمرکز ممکن است کارآمدتر باشد زیرا شبکه‌های بروز شده مستقیماً به تجهیزات لبه منتقل می‌شوند). تجهیزات لبه فاقد توان محاسباتی و ظرفیت ذخیره سازی داده مورد نیاز برای مسائل پیچیده هستند. چالش دیگردر طراحی گره های لبه، هزینه های مرتبط با انرژی و سخت افزار است.

اخیرا، تکنیک‌های یادگیری ماشین کاربردهای مفیدی در زمینه‌های تحقیقاتی مختلف و به‌ویژه در سیستم‌های محاسباتی تعبیه‌شده پیدا کرده‌اند. بررسی ها نشان می دهد[۳۷] که از بین تمامی تکنیک های یادگیری ماشین موجود، SVM، GMM، DNN، k-NN، HMM، درخت های تصمیم گیری، رگرسیون لجستیک، k-means و Bayes ساده تکنیک های رایجی هستند که برای برنامه های کاربردی تلفن همراه به کار می روند. Bayes ساده و درخت تصمیم از نظر هزینه های محاسباتی و حافظه پیچیدگی کمی دارند و بنابراین به بهینه سازی های نوآورانه نیاز ندارند. الگوریتم‌های رگرسیون لجستیک از نظر محاسباتی کم هزینه تر از Bayes و درخت‌های تصمیم ساده هستند، به این معنی که پیچیدگی کمتری دارند[۳۸]. HMM، K-NN، SVM، GMM، وDNN به هر حال از نظر محاسباتی و حافظه، فشرده هستند و از این رو، برای فشرده سازی کارآمد در محیطهای با منابع محدود، به تکنیکهای بهینه سازی جدید نیاز دارند لذا این الگوریتم‌ها ممکن است به طور موثر در محیط‌های محدود به منابع پیاده‌سازی شوند.

۴. یادگیری ماشین و محدویت منابع در گره‌های لبه

سیستم‌های محاسباتی تعبیه‌شده معمولاً از نظر توان محاسباتی و حافظه مورد نیاز محدود هستند. علاوه بر این، توان بسیار کم مصرف کنند و محدودیت های زمان واقعی را برآورده کنند. بنابراین، برای اینکه این مدل های یادگیری ماشین فشرده محاسباتی به طور موثر در فضای سیستم های تعبیه‌شده اجرا شوند، بهینه سازی های مناسب هم در معماری سخت افزار و هم در سطوح الگوریتم مورد نیاز است. در ادامه، روش‌های بهینه‌سازی را برای مقابله با تنگناها از نظر توان مصرفی، حافظه مصرفی و میزان تاخیر بررسی می‌کنیم.

تکنیک های یادگیری ماشین در حال حاضر محیط های با منابع محدود مانند دستگاه های تلفن همراه، دستگاه های تعبیه‌شده و سایر دستگاه های اینترنت اشیا را هدف قرار داده اند. لذا مروری بر پیشینه محیط‌های مختلف با محدودیت منابع مانند واحدهای میکروکنترلر(MCU) و دستگاه‌های تلفن همراه ارائه می‌کنیم. همچنین گزینه واحدهای شتاب سخت افزاری مورد استفاده برای سرعت بخشیدن به اجرای این الگوریتم ها در محیط های با منابع محدود را مورد بحث قرار می دهیم. بطور کلی میکروکنترلرها در سخت افزار، برای پیاده سازی سیستم های تعبیه‌شده متنوع و سایر برنامه های کاربردی اینترنت اشیا قرار دارند[۳۹]. یک میکروکنترلرشامل یک ریزپردازنده، حافظه، پورت های ورودی/خروجی و سایر تجهیزات جانبی است که همگی در یک تراشه ادغام شده اند. در هسته پردازنده میکروکنترلرها، پردازنده های همه منظوره برای محاسبات قرار دارند. می‌توانیم محدودیت منابع اکثر میکروکنترلرها را از نظر توان پردازشی موجود و حافظه روی تراشه مشاهده کنیم. این محدودیت منابع یک اشکال اساسی در پیاده سازی مدل های یادگیری ماشین است لذا تکنیک‌های فشرده‌سازی را بررسی می‌کنیم تا در حافظه موجود میکروکنترلر قرارگرفته تا محاسبات به‌طور کارآمد انجام شود.

تکنیک‌های یادگیری ماشین به تدریج در دستگاه‌های تلفن همراه برای کاربردهایی مانند تشخیص گفتار، بینایی رایانه، و غیره نفوذ می‌کنند. دستگاه‌های تلفن همراه ممکن است به دلیل منابع محاسباتی و حافظه محدود، تحت سیستم‌های محدود به منابع طبقه‌بندی شوند. از این رو، برای اجرای موفقیت آمیز الگوریتم های یادگیری ماشین در این دستگاه ها، بهینه سازی های مناسب باید انجام شود. در [۴۰] یک شتاب دهنده نرم افزاری برای تسریع در اجرای مدل های یادگیری در دستگاه های تلفن همراه توسعه داده شده است.

استنباط یادگیری ماشین در لبه به ویژه در میکروکنترلرهای بسیار کم توان در حال افزایش علاقه در بین جامعه الگوریتمهای یادگیری ماشین است. این علاقه به ایجاد یک پلتفرم مناسب می‌پردازد که در آن مدل‌های یادگیری ماشین ممکن است به طور موثر در دستگاه‌های IoT اجرا شوند. بنابراین، این یک حوزه تحقیقاتی رو به رشد در یادگیری ماشین سیستمهای تعبیه‌شده شده به نام TinyML باز کرده است. TinyML یک تکنیک یادگیری ماشین است که یادگیری ماشین فشرده و بهینه شده را برای سازگاری با میکروکنترلرهای بسیار کم مصرف یکپارچه می کند و اساساً با یادگیری ماشین ابری که در آن مدل‌های فشرده محاسباتی با استفاده از رایانه‌های پیشرفته در مراکز داده بزرگ پیاده‌سازی می‌شوند. TinyML پلتفرمی ایجاد می کند که به موجب آن مدل های یادگیری ماشین به دستگاه های کاربر منتقل می شوند تا تجربه کاربری خوب را برای برنامه های مختلف ارائه دهند و دارای مزایایی مانند بهره وری انرژی، کاهش هزینه ها، امنیت داده ها، تأخیر کم و غیره است که نگرانی های اصلی در ابر معاصر است. پلتفرم های مختلفی برای انتقال آسان الگوریتم های یادگیری ماشین به محیط های با محدودیت منابع توسعه یافته اند که معمولاً برای انتقال مدل‌های یادگیری ماشین به دستگاه‌های مختلف با منابع محدود سازگار استفاده می‌شوند.

۵. یادگیری ماشین و تشخیص ناهنجاری

به طور کلی، تشخیص ناهنجاری نقش مهمی در سیستم های نرم افزاری توزیع شده ایفا می کند. می تواند ارتباطات پیرامون رفتار سیستم را بهبود بخشد، تجزیه و تحلیل علت ریشه ای را بهبود بخشد و تهدیدات را برای اکوسیستم نرم افزار کاهش دهد. تشخیص ناهنجاری سنتی دستی است. با این حال، تکنیک های یادگیری ماشین موفقیت آشکارسازهای ناهنجاری را بهبود می بخشد. تشخیص ناهنجاری هر فرآیندی است که نقاط نامرتبط یک مجموعه داده را پیدا کند. این بدان معنی است که مواردی را که به مجموعه داده مرجع تعلق ندارند شناسایی می کند. این ناهنجاری ها ممکن است به ترافیک غیرعادی شبکه اشاره کنند. در دنیای امروزی سیستم های توزیع شده، مدیریت و نظارت بر عملکرد سیستم یک وظیفه اساسی برای دستیابی است. با وجود هزاران مورد برای نظارت، تشخیص ناهنجاری می تواند به نشان دادن محل وقوع خطا کمک کند و به سرعت پشتیبانی فنی را در مورد این مشکل دریافت کند. بنابراین، یادگیری ماشین با هدف مهندسی برای ایجاد یک سیستم تشخیص ناهنجاری که بهتر کار می کند، مجموعه داده های بزرگ را مدیریت می کند، تطبیق می دهد و به سرعت عمل می کند، مناسب است. استفاده از یادگیری ماشین برای تشخیص ناهنجاری مستلزم درک خوب مسئله است. تشخیص ناهنجاری از مقادیر زیادی داده سود می برد زیرا فرض بر این است که ناهنجاری ها نادر هستند. در این مقاله داده‌های آموزشی با «عادی» و «غیرعادی» برچسب‌گذاری شده‌اند. در یادگیری ماشین نظارت شده یک مجموعه داده به طور منظم با نقاط داده مشروح با برچسب آماده شده است. در این حالت، تمام نقاط ناهنجار از قبل مشخص شده اند. در سیستم های اینترنت اشیا، تشخیص ناهنجاری به عنوان تشخیص نفوذ شناخته می شود. در بیشتر موارد، نفوذ منجر به از بین رفتن محرمانه بودن و یکپارچگی، ازکارافتادگی و یا استفاده غیرمجاز از منابع می شود. برخی از نمونه‌های نفوذ رایج به شرح زیر است:

- تغییرات غیرمجاز در فایل‌های سیستم
- دسترسی غیرمجاز به داده ها و فایل‌های کاربر یا تغییر آنها
- تغییرات جداول یا سایر اطلاعات سیستم (شامل اطلاعات شبکه)
- استفاده غیرمجاز از منابع محاسباتی (ایجاد حساب‌های کاربری غیرمجاز)

به طور کلی، راه حل های پیشرفته بسیاری برای شناسایی حملات سایبری اینترنت اشیا با استفاده از رویکردهای هوش مصنوعی وجود دارد. اخیراً رویکردهای یادگیری ماشین مختلفی برای توسعه این رویکردها برای دستگاه‌های اینترنت اشیاء استفاده شده است. رفتار ترافیک عادی و ترافیک غیرعادی را می توان با دقت تشخیص نسبتاً بالا و نرخ مثبت کاذب پایین شناسایی کرد، بنابراین، امنیت داده های محرمانه و دستگاه های اینترنت اشیا را به حداکثر می رساند. به عنوان مثال، یک رویکرد جدید مبتنی بر ناهنجاری برای شبکه‌های اینترنت اشیا با یک موتور انتخاب ویژگی ترکیبی اجرا می‌شود که فقط مرتبط‌ترین ویژگی‌ها را انتخاب می‌کند و الگوریتم جنگل تصادفی هر ترافیک را به‌عنوان عادی یا غیرعادی طبقه‌بندی می‌کند. در این مقاله با استفاده از تکنیک های یادگیری ماشین، کارائی و دقت یک سیستم تشخیص نفوذ را بررسی می کنیم. هدف استفاده از مدل‌های طبقه‌بندی مختلف است که رفتارهای عادی و غیرعادی را تشخیص می‌دهد. مجموعه داده ای را انتخاب کرده و پیش پردازش مورد نیاز برای سازماندهی مجموعه داده را با انتخاب مهم ترین ویژگی ها و

آماده سازی آن برای ساخت مدل اعمال می کنیم. به دنبال این فرآیندها، الگوریتم‌های آموزشی مختلفی را اعمال می‌کنیم و بهترین طبقه‌بندی کننده یادگیری ماشین را ارزیابی می‌کنیم.

۱.۵. مجموعه داده

این بخش پیاده سازی مدل یادگیری ماشین را برای تشخیص ناهنجاری در حوزه اینترنت اشیا مورد بحث قرار می دهد. اسکریپت به زبان پایتون با استفاده از نوت بوک Google Collab نوشته شده است. کتابخانه یادگیری ماشین مورد استفاده Scikit-Learn است. این کتابخانه به طور گسترده در تمام مراحل اجرا برای طبقه‌بندی، تجزیه و تحلیل پیش‌بینی‌کننده و وظایف یادگیری ماشین استفاده می‌شود. تصمیم گرفتیم مدل طبقه بندی باینری را اعمال کنیم. طبقه بندی باینری به پیش بینی یکی از دو کلاس اشاره دارد. کلاس صفر نشان می دهد که رفتار عادی است در حالی که کلاس یک نشان می دهد که رفتار غیرعادی است. مجموعه داده KDD [۴۱] برای ارزیابی تشخیص نفوذ استفاده می شود که شامل طیف گسترده‌ای از نفوذهای شبیه‌سازی شده در یک محیط شبکه است. ابعاد مجموعه داده شامل ۴۹۴۰۲۱ ردیف × ۴۲ ستون است که بر روی آن پیش پردازش به جهت پیاده سازی بر روی تجهیزات لبه اینترنت اشیاء انجام می دهیم. پیش پردازش داده ها یک مرحله ضروری در یادگیری ماشین است زیرا کیفیت داده ها و اطلاعات مفیدی که می توان از آن به دست آورد مستقیماً بر توانایی مدل برای یادگیری تأثیر می گذارد. بنابراین، بسیار مهم است که داده ها را قبل از وارد کردن به مدل، پیش پردازش کنیم. ستون برچسب ها دارای ۲۳ نوع است، بنابراین اولین تغییری که انجام می دهیم این است که انواع آن را به '۰' (رفتار عادی) و '۱' (رفتار غیرعادی) تبدیل می کنیم. از آنجایی که هدف اعمال یک طبقه‌بندی باینری است، باید دو برچسب کلاس را به‌عنوان ۰ یا ۱ اختصاص دهیم. به جهت سادگی تصمیم گرفتیم یک طبقه‌بند باینری اعمال کنیم و نه یک طبقه‌بندی چند کلاسه. با داشتن مقادیر باینری در ستون برچسب، مدل آموزش‌دیده ۰ یا ۱ را پیش‌بینی می‌کند. پس از بررسی کل مجموعه داده، ویژگی‌های طبقه‌بندی را تجزیه و تحلیل می‌کنیم. بسیاری از الگوریتم‌های یادگیری ماشین انتظار داده‌های ورودی عددی را دارند زیرا نمی‌توانند مستقیماً روی داده‌های برچسب کار کنند. لذا باید داده های دسته بندی خود را به شکل عددی تبدیل کنیم. این رمزگذاری یک متغیر باینری جدید برای هر عدد صحیح منحصر به فرد اضافه می کند. کارایی الگوریتم ها با متغیرهای ورودی بیش از حد، کاهش می یابد. بنابراین با افزایش ابعاد، تعداد نقاط داده مورد نیاز برای عملکرد خوب به طور تصاعدی افزایش می یابد. قدرت پیش بینی هر طبقه بندی کننده با افزایش تعداد ابعاد افزایش می یابد، اما پس از پیشی گرفتن از تعداد مشخصی از ابعاد، عملکرد بدتر می شود. بنابراین باید کاهش ابعاد را اعمال کنیم. این فرآیند به تکنیک هایی اشاره دارد که تعداد متغیرهای ورودی را در یک مجموعه داده کاهش می دهد. هنگامی که با داده‌های با ابعاد بالا سروکار داریم، اغلب مفید است که ابعاد را با نمایش داده‌ها به یک زیرفضای با ابعاد پایین‌تر که ماهیت داده‌ها را در بر می‌گیرد، کاهش دهیم. این تکنیک در یادگیری ماشین برای ساده‌سازی مجموعه داده‌های طبقه‌بندی برای تطبیق بهتر با یک مدل پیش‌بینی استفاده می‌شود. ویژگی ها را با استفاده از انتخاب ویژگی تک متغیره انتخاب می کنیم.

۲.۵. آموزش و ارزیابی مدل

در این مرحله الگوریتم یادگیری ماشین با تغذیه مجموعه داده ها آموزش داده می شود. مدل آموزشی شامل داده‌های خروجی نمونه و مجموعه‌های مربوطه از داده‌های ورودی است که بر خروجی تأثیر دارند. مدل آموزشی برای اجرای داده های ورودی از طریق الگوریتم به منظور همبستگی خروجی پردازش شده با خروجی نمونه استفاده می شود. آموزش مدل در زبان ماشین فرآیند تغذیه الگوریتم مورد نظر با داده ها برای کمک به شناسایی و یادگیری مقادیر خوب برای همه ویژگی های استفده شده است. ۴ الگوریتم درخت تصمیم، k-نزدیکترین همسایه، رگرسیون لجستیک و جنگل تصادفی را انتخاب کرده ایم. درخت تصمیم از برخی توابع هزینه برای انتخاب بهترین طبقه بندی استفاده می کند. لذا سعی می کنیم بهترین ویژگی را پیدا کنیم که بهترین عملکرد را در طبقه بندی داده های آموزشی داشته باشد. این فرآیند تا رسیدن به یک گره برگ تکرار می شود و بنابراین به آن طبقه بندی باینری بازگشتی می گویند. مشهود است که در درخت تصمیم ریشه در بالا قرار دارد و این ریشه به چندین گره مختلف تقسیم می شود. هدف یادگیری ماشین کاهش عدم قطعیت در مجموعه داده است، بنابراین مفاهیمی مانند آنتروپی و شاخص جینی را استفاده می کنیم. ایده اصلی درخت تصمیم، شناسایی ویژگی هایی است که حاوی بیشترین اطلاعات برای ویژگی هدف هستند و سپس مجموعه داده را در امتداد مقادیر ویژگی ها به گونه ای

تفکیک می کند که خروجی ویژگی هدف تا حد امکان خالص باشد. حرکت به سمت پایین منجر به کاهش سطح عدم قطعیت می شود و طبقه بندی بهتری را در هر گره انجام می دهد. آنتروپی عدم قطعیت را در مجموعه داده تعریف می کند. معادله آنتروپی در طبقه بندی به صورت زیر است:

$$Entropy = -\sum_{i=1}^{n} p_i * log_2(p_i) \quad (۱)$$

آنتروپی اساساً ناخالصی یک گره را اندازه گیری می کند. ناخالصی درجه تصادفی است که نشان می دهد داده‌ها چقدر تصادفی هستند. برای مثال، آنتروپی در یک موقعیت خاص، بدون دانستن رفتار آنتروپی در گره والد محاسبه می شود. آنتروپی نقش مهمی در محاسبه بهره اطلاعات ایفا می کند. اطلاعات بدست آمده برای تعیین کمیت اینکه کدام ویژگی حداکثر اطلاعات را در مورد طبقه بندی بر اساس مفهوم آنتروپی ارائه می دهد، اعمال می شود. این کار با محاسبه عدم قطعیت، بی نظمی یا ناخالصی، با هدف کاهش مقدار آنتروپی از بالا (گره ریشه) به پایین (گره های برگ) انجام می شود:

$$Gain(T, X) = Entropy(T) - Entropy(T, X) \quad (۲)$$

شاخص جینی یا اندازه گیری ناخالصی جینی یکی از روش هایی است که در الگوریتم های درخت تصمیم برای تصمیم گیری تقسیم بهینه از یک گره ریشه و از تقسیم های بعدی استفاده می شود. کاهش شاخص جینی احتمال طبقه بندی اشتباه را کاهش می دهد:

$$GiniIndex = 1 - \sum_{i=1}^{n} (p_i)^2 \quad (۳)$$

برای تنظیم مدل پارامترهای مهمی در درخت تصمیم وجود دارد که بر مدل از نظر بیش برازش و کم‌برازش تأثیر می‌گذارند.

۳.۵. نتایج و ارزیابی مدل

ابعاد نهایی مجموعه داده ۴۹۴۰۲۱ سطر و ۱۱ ستون است. مجموعه داده را به یک مجموعه آموزشی (۷۵٪) و یک مجموعه آزمایشی (۲۵٪) تفکیک کردیم. مجموعه آموزشی مجموعه ای است که به مدل تغذیه می شود در حالی که مجموعه تست مجموعه ای است که امکان ارزیابی عملکرد مدل را فراهم می کند. برای بررسی عملکرد مدل، معیارهای مختلف عملکرد مانند صحت، دقت، فراخوانی و امتیاز F1را محاسبه می‌کنیم.

- **صحت،** تعداد نقاط داده به درستی پیش بینی شده از تمام نقاط داده است:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (۴)$$

- **دقت،** نسبت نقاط داده مثبت پیش بینی شده صحیح به کل نقاط داده مثبت پیش بینی شده است:

$$Precision = \frac{TP}{TP + FP} \quad (۵)$$

- **فراخوانی (حساسیت)،** نسبت نقاط داده مثبت پیش بینی شده صحیح به تمام نقاط داده در کلاس واقعی است:

$$Precision = \frac{TP}{TP + FP} \quad (۶)$$

- **امتیازF1** ، میانگین وزنی دقت و فراخوانی است:

$$Recall = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \quad (۷)$$

ماتریس سردرگمی یک ماتریس NxN است که برای ارزیابی عملکرد یک مدل طبقه‌بندی استفاده می‌شود، که در آن N تعداد کلاس‌های هدف است و این ماتریس، مقادیر هدف واقعی را با مقادیر پیش‌بینی شده توسط مدل مقایسه می کند (شکل ۱).

Actual Values

Predicted Values	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

	درخت تصمیم	
	عادی	غیرعادی
عادی	۱۹۲۹۰	۱۳۲
غیرعادی	۵۵۶	۷۸۸۴۷

	رگرسیون لجستیک	
	عادی	غیرعادی
عادی	۱۹۰۷۲	۳۵۰
غیرعادی	۱۰۴۶	۷۸۳۵۷

	K-نزدیکترین همسایه	
	عادی	غیرعادی
عادی	۱۹۲۱۲	۲۱۰
غیرعادی	۶۲۲	۷۸۷۸۱

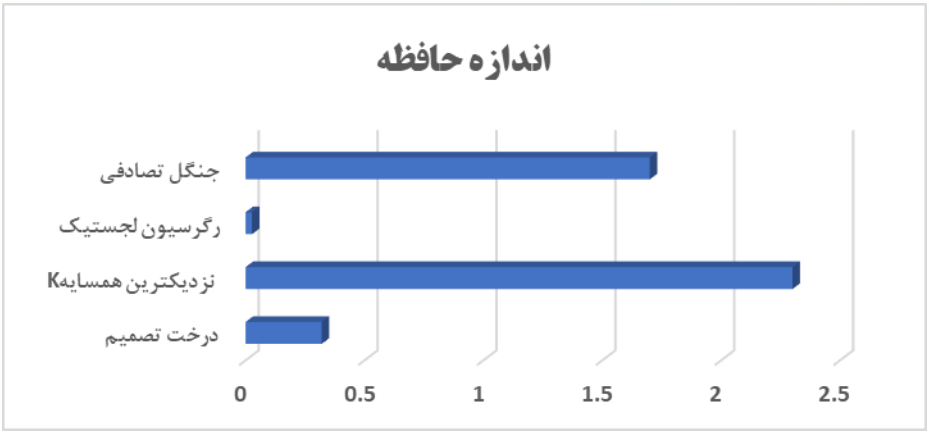
	جنگل تصادفی	
	عادی	غیرعادی
عادی	۱۹۲۹۲	۱۳۰
غیرعادی	۵۴۰	۷۸۸۶۳

شکل ۱. جداول ماتریس سردرگمی الگوریتمهای یادگیری ماشین

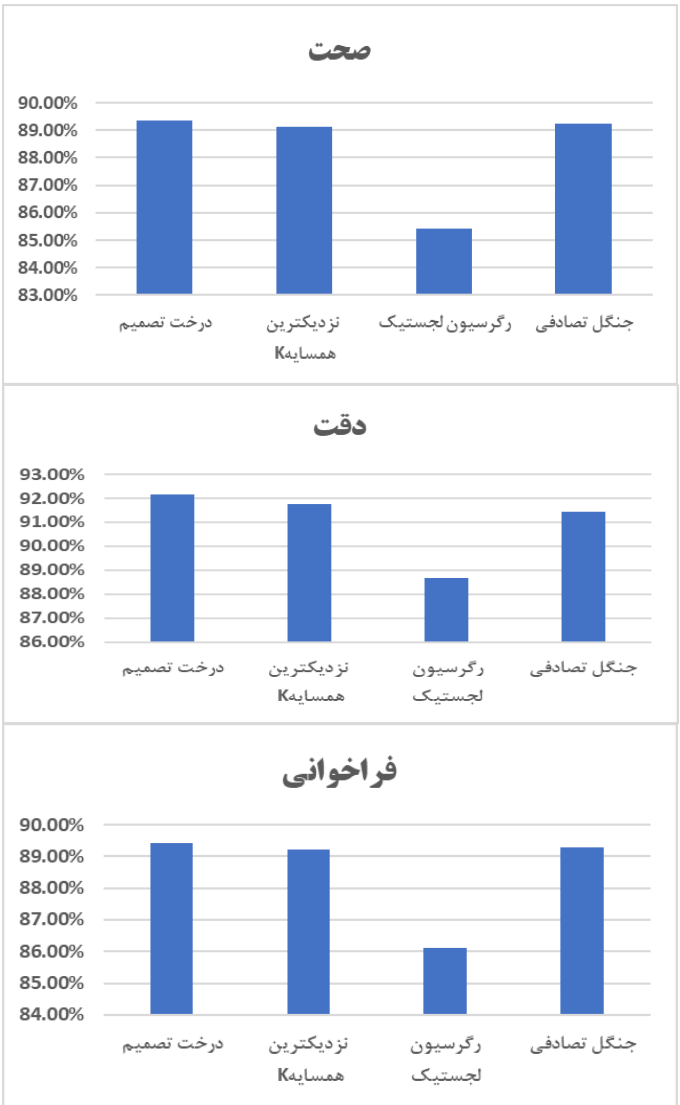
در جدول۱ قابل مشاهده است که درخت تصمیم به بالاترین دقت ۹۲.۱۷% دست یافته است. برای پیاده سازی آن بر روی دستگاه لبه باید مدلی را انتخاب کنیم و لازمه انتخاب مدل داشتن دقت بالا، پیچیدگی زمان اجرا سریع و میزان حافظه مصرفی است. لذا، درخت تصمیم انتخاب می شود. نمودار مقایسه ای میزان حافظه مصرفی مدل های آموزش داده شده در شکل ۲ نشان داده شده است. همچنین نمودارهای ارزیابی معیارهای صحت، دقت، فراخوانی و امتیاز F1 برای هریک از مدل های آموزش داده شده بر روی داده های آزمایش در شکل ۳ نشان داده شده است. با بررسی پارامترهای معیارهای ارزیابی، زمان اجرای مدل و میزان حافظه مصرفی، درخت تصمیم برای پیاده سازی بر روی گره لبه انتخاب می شود.

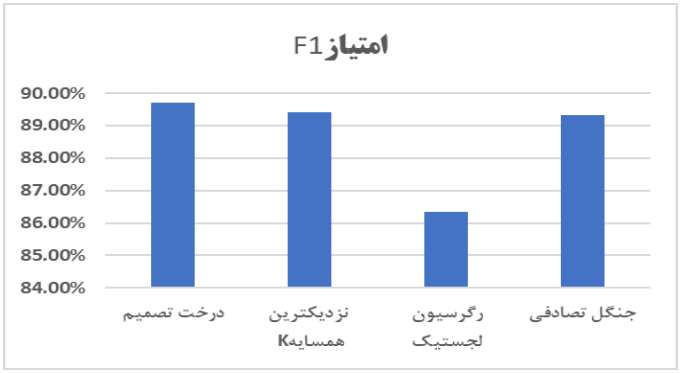
جدول ۱. معیارهای ارزیابی الگوریتمهای یادگیری ماشین

الگوریتم یادگیری ماشین	صحت	دقت	فراخوانی	امتیاز F1	اندازه حافظه
درخت تصمیم	٪۸۹.۳۵	٪۹۲.۱۷	٪۸۹.۴۲	٪۸۹.۷۲	۳۲۰KB
k-نزدیکترین همسایه	٪۸۹.۱۱	٪۹۱.۷۵	٪۸۹.۲۲	٪۸۹.۴۱	۲.۳MB
رگرسیون لجستیک	٪۸۵.۴۲	٪۸۸.۶۷	٪۸۶.۱۲	٪۸۶.۳۴	۲۷KB
جنگل تصادفی	٪۸۹.۲۲	٪۹۱.۴۳	٪۸۹.۲۷	٪۸۹.۳۱	۱.۷MB



شکل ۲. نمودار مقایسه ای میزان حافظه مصرفی مدل های آموزش داده شده





شکل ۳. نمودار ارزیابی معیارهای صحت، دقت، فراخوانی و امتیاز F1 مدل های آموزش داده شده بر روی داده های آزمایش

۶. نتیجه گیری

در این مقاله، تشخیص ناهنجاری را برای پیش بینی حملات سایبری برای اینترنت اشیا با استفاده از یادگیری ماشین مدل‌سازی کردیم. با توجه به توسعه سریع سیستمهای اینترنت اشیا در حوزه های مختلف، مقادیر زیادی داده به طور مداوم در حال تولید است که نیاز به تمرکز بیشتر بر حریم خصوصی و امنیت دارد. حملات در اینترنت اشیا شامل انواع مختلفی است. اگر این نوع حملات موفقیت آمیز باشد، عملکرد اینترنت اشیا می تواند از طرق مختلفی مانند ارائه اطلاعات نادرست به خطر بیفتد. امروزه رویکرد هوش مصنوعی مزایای بسیاری دارد که می‌توان آن را یکی از امیدوارکننده‌ترین روش‌ها دانست. بنابراین طبقه‌بندی باینری را مدل‌سازی کردیم که پیش‌بینی یکی از دو کلاس (رفتار عادی و رفتارغیرعادی) در مجموعه داده KDD انجام شد. الگوریتم های مختلف یادگیری ماشین مانند درخت تصمیم، k-نزدیک ترین همسایه، رگرسیون لجستیک و جنگل تصادفی مدل‌سازی شدند. درخت تصمیم را به عنوان بهترین مدل با عملکرد مناسب با توجه به مبادله بهینه بین دقت (۹۲.۱۷٪) و میزان حافظه مصرفی و همچنین پیچیدگی محاسباتی انتخاب نمودیم. با ارزیابی رفتار در یک سناریوی واقعی، بسط بیشتر به استقرار مدل یادگیری ماشین انتخاب شده در دستگاه لبه خواهد بود.

منابع

1. Wayne, W. Praise of High-Performance Embedded Computing: Architectures, Applications, and Methodologies; Morgan Kaupmann Publishers: CA, USA, 2007.
2. Haigh, K.Z.; Mackay, A.M.; Cook, M.R.; Lin, L.G. Machine Learning for Embedded Systems: A Case Study; BBN Technologies: MA, USA, 2015; Volume 8571, pp. 1–12.
3. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks Alex. Adv. Neural Inf. Process. Syst. 2012, 25, 1097–1105.
4. Szegedy, C.; Liu, W.; Jia, P.Y.; Reed, S.S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA, 2015; pp. 1–9.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference Computer Vision Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
6. Real, E.; Moore, S.; Selle, A.; Saxena, S.; Suematsu, Y.L.; Tan, J.; Le, Q.V.; Kurakin, A. Large-scale evolution of image classifiers. In Proceedings of the 34th International Conference Machine Learning ICML, Australia, 6–11 August 2017; pp. 4429–4446.
7. Tan, M.; Le, Q.V. Efficient Net: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference Machine Learning ICML 2019, Long Beach, CA, USA, 10–15 June 2019; pp. 10691–10700.

8. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal. Process. Mag. 2012, 29, 82–97.
9. Chan,W.; Jaitly, N.; Le, Q.V.; Vinyals, O. Listen, attend and spell. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016.
10. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey,W. ; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's neural machine translation system: Bridging the Gap between human and machine translation. arXiv 2016.
11. Collobert, R.;Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. J. Mach. Learn. Res. 2011, 12, 2493–2537.
12. Haj, R.B.; Orfanidis, C. A discreet wearable long-range emergency system based on embedded machine learning. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications Workshops, Kassel, Germany, 22–26 March 2021.
13. Dean, J. The deep learning revolution and its implications for computer architecture and chip design. In Proceedings of the 2020 IEEE International Solid-State Circuits Conference-(ISSCC), San Francisco, CA, USA, 16–20 February 2020; pp. 8–14.
14. Khan, M.A.; Kim, J. toward developing efficient Conv-AE-based intrusion detection system using heterogeneous dataset. Electronics 2020, 9, 1771.
15. Li, P.; Luo, Y.; Zhang, N.; Cao, Y. Hetero Spark: A heterogeneous CPU/GPU spark platform for machine learning algorithms. In Proceedings of the 2015 IEEE International Conference Networking, Architecture Storage, NAS, Boston, MA, USA, 6–7 August2015; pp. 347–348.
16. Raparti, V.Y.; Pasricha, S. RAPID: Memory-aware NoC for latency optimized GPGPU architectures. IEEE Trans. Multi-Scale Computer. Syst. 2018, 4, 874–887.
17. Cheng, X.; Zhao, Y.; Robaei, M.; Jiang, B.; Zhao, H.; Fang, J. A low-cost and energy-efficient noc architecture for GPGPUs. J. Nat. Gas Geosci. 2019, 4, 1–28.
18. Zhang, L.; Cheng, X.; Zhao, H.; Mohanty, S.P.; Fang, J. Exploration of system configuration in effective training of CNNs on GPGPUs. In Proceedings of the 2019 IEEE International Conference Consumer Electronics ICCE, Las Vegas, NJ, USA, 11 January2019; pp. 1–4.
19. Cui, X.; Liu, H.; Fan, M.; Ai, B.; Ma, D.; Yang, F. Seafloor habitat mapping using multibeam bathymetric and backscatter intensity multi-features SVM classification framework. Appl. 2020, 174, 107728.
20. Yu, Q.; Wang, C.; Ma, X.; Li, X.; Zhou, X. A deep learning prediction process accelerator-based FPGA. In Proceedings of the 2015 IEEE/ACM 15th International Symposium Cluster Cloud, Grid Computer CC Grid 2015, China, 4–7 May 2015; pp. 1159–1162.
21. Noronha, D.H.; Zhao, R.; Goeders, J.; Luk, W.; Wilton, S.J.E. On-chip FPGA debug instrumentation for machine learning applications. In Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Seaside, CA, USA, 24–26 February 2019.
22. Wang, C.; Gong, L.; Yu, Q.; Li, X.; Xie, Y.; Zhou, X. DLAU: A scalable deep learning accelerator unit on FPGA. IEEE Trans. Computer. Des. Integrated. Circuits Syst. 2016, 513–517.
23. Chang, A.X.M.; Martini, B.; Culurciello, E. Recurrent Neural Networks Hardware Implementation on FPGA.
24. Branco, S.; Ferreira, A.G.; Cabral, J. Machine learning in resource-scarce embedded systems, FPGAs, and end-devices: A survey. Electronics 2019, 8, 1289.
25. Zhang, C.; Li, P.; Sun, G.; Guan, Y.; Xiao, B.; Cong, J. Optimizing FPGA-based accelerator design for deep convolutional neural networks. In Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, USA, 2015; pp. 161–170.

26. Neshatpour, K.; Mokrani, H.M.; Sasan, A.; Ghasemzadeh, H.; Rafatirad, S.; Homayoun, H. Architectural considerations for FPGA acceleration of machine learning applications in MapReduce. In Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, Greece, 15–19 July 2018.
27. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. Squeeze Net: Alex Net-level Accuracy With 50 Fewer Parameters and <0.5 mb Model Size.
28. Deng, Y. Deep learning on mobile devices: A review. In Proceedings of the SPIE 10993, Mobile Multimedia/Image Processing, Security, and Applications 2019, 109930A, Baltimore, ML, USA, 14–18 April 2019.
29. Kim, D.; Ahn, J.; Yoo, S. A novel zero weight/activation-aware hardware architecture of convolutional neural network. In Proceedings of the 2017 Design, Automation and Test in Europe DATE 2017, Switzerland, 27–31 March 2017; pp. 1462–1467.
30. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. Nature 2015, 521, 436–444.
31. Schmidhuber, J. Deep learning in neural networks: Neural network. 2015, 61, 85–117.
32. Jawandhiya, P. Hardware design for machine learning. Int. J. Artif. Intell. Appl. 2018, 9, 1–6.
33. Chen, J.; Ran, X. Deep learning with edge computing: Proc. IEEE 2019, 107, 1655–1674.
34.Frank, M.; Drikakis, D.; Charissis, V. Machine-learning methods for computational science and engineering. Computation2020, 8, 15.
35. Xiong, Z.; Zhang, Y.; Niyato, D.; Deng, R.; Wang, P.; Wang, L.C. Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges. IEEE Veh. Technol. Mag. 2019, 14, 44–52.
36. Carbonell, J.G. Machine learning research. ACM SIGART Bull. 1981, 18, 9.
37. Jadhav, S.D.; Channe, H.P. Comparative STUDY of K-NN, naive bayes and decision tree classification techniques. Int. J. Sci. Res.2016, 5, 1842–1845.
38. Logistic Regression as a Classifier. December 2020.
39.Lane, N.D.; Bhattacharya, S.; Georgiev, P.; Forlivesi, C.; Jiao, L.; Qendro, L.; Kawsar, F. DeepX: A Software accelerator for low-power deep learning inference on mobile devices. In Proceedings of the 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Vienna, Austria, 11–14 April 2016.
40.Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016; pp. 1–14.
41.«KDD Cup 1999 Data». In: The UCI KDD Archive (1999).