سالنامه‌مخابرات (۲۰۲۵) ۸۰: ۲۸۳–۲۹۷
https://doi.org/10.1007/s12243-024-01041-5

TinyML جعبه ابزاری برای به روزرسانی های امن بی سیم و ارزیابی عملکردمدل های :RIOT-ML

ژائولان‌هوانگ[۱،۲]·کوئن زندبرگ[۱]·کاسپار شلیزر[۱]·امانوئل باچلی[۱،۲]

دریافت:۱۷ ژانویه ۲۰۲۴ / پذیرش: ۱۲ مه ۲۰۲۴ / انتشار آنلاین: ۲۲ مه ۲۰۲۴

چکیده

متخصصان‌حوزه TinyML تاکنون فاقد یک جعبه ابزار جامع و «شامل باتری» برای ساده سازی ادغام مداوم، استقرار مداوم و ارزیابی عملکرداجرای مدل های مختلف یادگیری ماشین بر روی سخت افزارهای مختلف اینترنت اشیا کم مصرف بوده اند. مقاله ما با پرداختن به‌این شکاف، RIOT-ML را معرفی می کند، یک جعبه ابزار همه کاره که برای کمک به طراحان و محققان اینترنت اشیا در این وظایف ساخته‌شده است. برای این منظور، ما RIOT-ML را بر اساس ادغام مجموعه ای از قابلیت ها از یک سیستم عامل تعبیه شده کم مصرف،یک ترنسپایلر و کامپایلر مدل جهانی، یک جعبه ابزار برای اندازه گیری عملکرد TinyML و یک چارچوب به روزرسانی امن کم مصرف‌از طریق هوا طراحی کردیم - که همه آنها در یک بستر آزمایشی اینترنت اشیا با دسترسی آزاد در دسترس جامعه قابل استفاده هستند.پیاده سازی متن باز ما از RIOT-ML و آزمایش های اولیه ای که در مورد آنها گزارش می دهیم، کاربرد آن را در ارزیابی تجربی عملکردمدل TinyML در میان ناوگان بردهای اینترنت اشیا کم مصرف تحت آزمایش در میدان، با طیف گسترده ای از معماری های میکروکنترلرناهمگن و پیکربندی های اتصال شبکه ناوگان، نشان می دهد. وجود یک جعبه ابزار متن باز مانند RIOT-ML برای تسریع تحقیقات‌در زمینه ترکیب هوش مصنوعی و اینترنت اشیا و تقویت تحقق کامل پتانسیل محاسبات لبه ضروری است.

کلمات‌کلیدیهوش مصنوعی·اینترنت اشیا·یادگیری ماشین·قدرت کم·میکروکنترلر·معیارها·به روزرسانی نرم افزار·MLOps

امقدمه

همچنان‌که هوش مصنوعی (AI) بیشتر و بیشتر در زندگی ما نفوذ می کند،مکانیسم هایی مانند شبکه های عصبی عمیق [۱] در مکان های بیشترو بیشتری در سیستم های توزیع شده مختلف مورد استفاده قرار می گیرند(یا استقرار آنها برنامه ریزی شده است). به طور خاص، یک شبکه‌حسگر بی سیم (WSN) می تواند با استقرار هوش مصنوعی در گره های‌لبه، پوشش و اتصال خود را بهبود بخشد و مصرف انرژی و پهنای باندرا کاهش دهد [۲].

خط‌لوله داده با هوش مصنوعی معمولاً نیاز به ایجاد و استفاده از ...دارد.*مدل*، یعنی یک ساختار لایه ای از الگوریتم های پیچیده (که به‌عنوان ... نیز شناخته می شود)*اپراتورها*) که داده ها را تفسیر کرده و براساس آن داده ها تصمیم گیری می کنند. این مدل ابتدا باید آموزش داده‌شود (*یادگیری*فاز [۱])، قبل از اینکه بتواند به مرحله تولید برسد (مورداستفاده برای*استنباط*).

کارهای‌اخیر از یادگیری ماشین کوچک (TinyML) [۳، ۴] تلاش های‌جامعه برای بهینه سازی مدل ها برای تطبیق با بودجه های‌منابع کوچک تر (و در عین حال عملکرد کارآمد) میکروکنترلرهای‌کم مصرف در اینترنت اشیا (IoT). در نتیجه، هر دوامکان یادگیری و جایگذاری استنتاج برای در بر گرفتن ترمینال های‌فوق العاده کم مصرف گسترش می یابند.

بااین حال، ابزارهای متن باز عمومی و راحت، فاقد طراحانی هستند که بتوانندترکیبی از هوش مصنوعی و اینترنت اشیا (AIoT) را به کار گیرند، طراحانی‌که ملزم به انجام موارد زیر باشند:

- عملکردمدل های خود را هنگامی که در امتداد پیوستار ابر-لبه-ترمینال‌قرار می گیرند، ارزیابی کنند، به خصوص هنگامی که امکان‌قرارگیری آنها در دستگاه های مختلف مبتنی بر میکروکنترلردر نظر گرفته شود.
- مدل های‌خود را به دقت تنظیم کنند و تنگناهای عملکرد را در جزئیات‌لایه مدل، در میکروکنترلرهای مختلف شناسایی کنند.
- یک‌میکروکنترلر مناسب برای اجرای مدل خود، برای یک کار هدفمند که روی‌یک دستگاه کم مصرف که قرار است طراحی شود، اجرا می شود، انتخاب‌کنید.
- به روزرسانی‌مداوم و ایمن مدل های از پیش آماده شده روی ناوگانی‌از دستگاه های ناهمگن تحت آزمایش، از راه دور، از طریق‌شبکه
- نظارت‌بر عملکرد محاسباتی مدل های مستقر شده از راه دور، ازطریق شبکه، مثلاً روی داده های جدید جمع آوری شده در محل

ب ژائولان‌هوانگ
zhaolan.huang@fu-berlin.de

۱ دانشگاه‌آزاد برلین، برلین، آلمان اینریا ساکلای،

۲ پاریس،فرانسه

بنابراین،این مقاله به معرفی موارد زیر می پردازد:*جعبه ابزار RIOT برای یادگیری ماشینی*(.یک گردش کار را برای فشرده سازی،فلش کردن و ارزیابی خودکار مدل های دلخواه پیاده سازی‌می کند RIOT-ML .کم مصرف را ترکیب می کند IoT متن بازاست که به طور تنگاتنگی یک کامپایلر مدل عمومی و یک سیستم‌عامل محبوب AIoT یک جعبه ابزار ،)RIOT-ML[۱]روی بردهای‌کم مصرف تجاری (COTS) دلخواه[۲]بر اساس معماری های مختلف‌میکروکنترلر (MCU) محبوب. با بهره گیری از یک پشته شبکه‌کم مصرف و پرکاربرد در ترکیب با مکانیسم های نرم افزاری امن‌اینترنت اشیا، RIOT-ML همچنین امکان کنترل، نظارت و به روزرسانی‌مدل ML را روی مجموعه هایی از چنین دستگاه هایی ازراه دور، بر روی توپولوژی های شبکه ناهمگن فراهم می کند.

مشارکت های‌مقاله ایکمک های ما به شرح زیر است:

- مایک جعبه ابزار جهانی برای ارزیابی داخلی و نظارت از راه دور مدل های)RIOT-ML(TinyML روی دستگاه های کم مصرف طراحی‌کردیم. RIOT-ML جعبه ابزار U-TOE را ادغام و گسترش می دهد[۵]، بررسی های امکان سنجی برای استفاده از مدل های دلخواه‌در طیف گسترده ای از پلتفرم های سخت افزاری اینترنت اشیارا ارائه می دهد. این به محققان و توسعه دهندگان اجازه می دهدتا تنگنای عملکرد یک مدل مشخص را در یک دستگاه هدف‌پیدا کنند. نتایج ارزیابی، امکان طراحی مشترک با سایر اجزا درسطح سیستم را فراهم می کند و به بهینه سازی مدل ها و پیکربندی های‌ML برای موارد استفاده خاص کمک می کند و امکان‌دستیابی به بهترین عملکرد ممکن در دستگاه های هدف را فراهم‌می کند.
- مامکانیزم هایی را برای به روزرسانی های امن مدل از طریق هوا(OTA) و همچنین برای استقرار و مدیریت مداوم مدل های‌دلخواه بر روی دستگاه های با محدودیت منابع بر روی‌پیکربندی های شبکه دلخواه، که ممکن است شامل لینک های‌شبکه کم مصرف نیز باشد، طراحی، پیاده سازی و ارزیابی‌می کنیم.
- کدرا منتشر کردیم[۳]از RIOT-ML تحت مجوز متن باز. این پیاده سازی‌امکان کامپایل، فلش کردن، ارزیابی و به روزرسانی های‌OTA امن شبکه های عصبی مختلف (مدل های‌محاسباتی مبتنی بر گراف) را از چارچوب های اصلی یادگیری‌ماشین بر روی بردهای کم مصرف مختلف مبتنی بر معماری های‌مجموعه دستورالعمل محبوب (ESP32، RISC-V، ARM Cortex-M) فراهم می کند.
- مامعیارها و ارزیابی تجربی مقایسه ای را با استفاده از RIOT-ML ارائه‌می دهیم که هم در یک بستر آزمایشی اینترنت اشیا با دسترسی‌آزاد و هم در ایستگاه های کاری شخصی قابل تکرار است، که‌بینش هایی در مورد عملکرد استنتاج با مدل های مختلف روی سخت افزارهای‌کم مصرف مختلف ارائه می دهد و نشان می دهد که‌چگونه RIOT-ML می تواند توسط محققان و توسعه دهندگان تجربی‌TinyML برای تنظیم دقیق پیکربندی های اینترنت اشیا دوباره مورداستفاده قرار گیرد.

[۱] خروجی TensorFlow، PyTorch...

[۲] مانند BBC: microbit، nrf52840dk، Arduino Zero، HiFive...

[۳] دیدن https://github.com/TinyPART/RIOT-ML

۲کارهای مرتبط

کارهای‌اخیر بررسی کرده اند [۶] دامنه چارچوب ها، وظایف و معیارهای‌یادگیری ماشین، از جمله بررسی جامع پشته TinyML و خطلوله استقرار. برای تطبیق با بودجه های منابع ناچیز معمول میکروکنترلرها(کیلوبایت حافظه، مصرف برق بر حسب میلی وات،فرکانس واحد پردازش مرکزی (CPU) بر حسب مگاهرتز) و در عین‌حال حفظ عملکرد در سطح قابل قبول و حفظ قابلیت حمل به‌سخت افزارهای بسیار چندریختی در این دسته، باید با تعدادی ازچالش ها روبرو شد.

پلتفرم های‌نرم افزاری اینترنت اشیا تعبیه شدهسیستم های عامل متن بازاینترنت اشیا مختلفی برای ارائه انتزاع سخت افزار، اشتراک گذاری‌منابع اولیه و دسترسی راحت به لوازم جانبی (مانند حسگر/ محرک،زیرسیستم شبکه) روی میکروکنترلرهای ناهمگن کم مصرف استفاده‌می شوند. کارهای قبلی مانند [۷] چنین سیستم های عاملی را بررسی‌می کند، که از جمله نمونه های برجسته آن می توان به RIOT [اشاره‌کرد.[۸] و زفیر.[۴]با این حال، تاکنون چنین پلتفرم های نرم افزاری هیچ پشتیبانی‌از چارچوب یادگیری ماشین ارائه نمی دهند - یا اگر ارائه می دهند،این پشتیبانی بسیار محدود است. علاوه بر این، پیشرفته ترین پشتیبانی هاتاکنون معمولاً سخت افزاری هستند - یا مختص فروشنده، مثلاً با کتابخانه هایی که توسط STM32CubeMx یا ARM CMSIS-NN ارائه‌می شوند.

بسترهای‌آزمایشی اینترنت اشیا کم مصرفبسترهای آزمایشی مختلف، دسترسی‌از راه دور به مجموعه ای از دستگاه های مبتنی بر میکروکنترلر قابل برنامه ریزی‌مجدد را ارائه می دهند. کارهای قبلی مانند [۹،۱۰] چنین بسترهای آزمایشی‌را بررسی کنید، که از جمله نمونه های برجسته آن می توان به آزمایشگاه‌اینترنت اشیا با دسترسی آزاد اشاره کرد [۱۱]که دسترسی از راه دور Bare-Metal (سریال از طریق پروتکل کنترل انتقال (TCP)) را به ناوگانی متشکل‌از صدها برد کم مصرف محبوب از انواع مختلف ارائه می دهد.

مجموعه های‌بنچمارک برای TinyMLبنچمارک کردن یادگیری ماشینی‌روی سخت افزار کم مصرف، چالش های متعددی را به دنبال‌دارد [۱۲]کارهای قبلی مانند MLPerf Tiny [۱۳] یک مجموعه معیاراستاندارد (مجموعه ای ثابت از وظایف نماینده یادگیری ماشین) برای ارزیابی عملکرد سخت افزار معین و یک پلتفرم آنلاین‌برای تولیدکنندگان جهت انتشار نتایج معیار مقایسه ای خود ارائه‌می دهد. در مقابل، RIOT-ML یک جعبه ابزار قدرتمندتر و قابل‌تنظیم تر برای انجام بررسی های امکان سنجی مدل های یادگیری‌ماشین تعریف شده توسط کاربر در دستگاه های کم مصرف،با درجه انعطاف پذیری و سفارشی سازی بیشتر ارائه می دهد.

بنچمارک های‌TinyMLکارهای قبلی مانند [۱۴] بر مقایسه عملکردچارچوب های مختلف یادگیری ماشین روی دو برد کم مصرفCOTS (Arduino Nano BLE 33 و F401RE- STM32 NUCLEO) تمرکز دارد. به طور خاص،

[۴] ببینید https://www.zephyrproject.org/

دوچارچوب TinyML، Tensorflow Lite برای میکروکنترلرها (TFLM) وX-CUBE-AI را در زمینه تشخیص حرکت و تشخیص کلمه بیدارباش،محک زدند. کارهای دیگری مانند [۱۵]مدل های TFLM را روی‌چندین برد مبتنی بر میکروکنترلر آزمایش کردند. در حالی که چنین مقالاتی‌مقایسه عملکرد چارچوب های خاص را روی بردهای خاص برای‌وظایف خاص ارائه می دهند، RIOT-ML انعطاف پذیری و عمومیت‌بیشتری ارائه می دهد و به توسعه دهندگان اجازه می دهد طیف‌وسیع تری از مدل های (مشخص شده توسط کاربر) را روی انواع بیشتری‌از دستگاه های کم مصرف ارزیابی کنند و به جزئیات اجرای مدل هایML بپردازند.

کامپایلرو ترنسپایلر مدل TinyMLکامپایلرهایی مانند ماشین مجازی‌تنسور (TVM) [۱۶]می توان از آن برای خودکارسازی ترجمه و کامپایل‌مدل های ارائه شده توسط چارچوب های اصلی یادگیری ماشین(Pytorch، TFML و غیره) استفاده کرد تا روال های سطح پایین‌را در معرض نمایش قرار داده و آنها را برای اجرا بر روی ویژگی های‌خاص واحد پردازش (CPU)، واحد پردازش گرافیکی (GPU) و غیره) بهینه کند. اخیراً افزونه ای از TVM به نام uTVM معرفی‌شده است که اهداف سخت افزاری کوچکتری از جمله انواع MCUها (واحدهای میکروکنترلر) را اضافه می کند.

پروفایلرهای‌مدل TinyML[ML-EXray[۱۷] توسعه دهندگان TinyML را قادر می سازد تا به جزئیات سطح لایه اجرای ML دسترسی‌پیدا کنند و مشکلات استقرار ابری تا لبه را تشخیص دهند. توسعه دهندگان‌می توانند خطوط لوله استقرار لبه را با قابلیت استفاده‌بالا، با استفاده از کمتر از 15 خط کد برای بررسی کامل، تجزیه وتحلیل و اشکال زدایی کنند. با این حال، اتکا به Tensorflow Lite قابلیت‌تطبیق مدل ها از چارچوب های ML بیشتر و استقرار آنها در دستگاه های‌کم مصرف را محدود می کند. چارچوب های اصلی ML (MXNet، Pytorch، TFML و غیره) پروفایلر داخلی ارائه می دهند [۱۸]چنین ابزارهایی به توسعه دهندگان اجازه می دهند عملکرد مدل های‌خود را اندازه گیری کنند. می توان از آن ها برای جمع آوری معیارهایی‌مانند زمان استنتاج و میزان استفاده از حافظه استفاده کرد که‌سپس می توان آن ها را برای بهینه سازی عملکرد مدل تجزیه و تحلیل‌کرد. اگرچه می تواند جزئیات اجرا را در سطح لایه در اختیار ما قراردهد، اما هنوز فاقد پشتیبانی از استقرار و ارزیابی روی دستگاه در دستگاه های‌مختلف اینترنت اشیا است، در حالی که RIOT-ML یک جعبه‌ابزار عمومی تر است که یک راه حل جامع در طیف وسیعی از دستگاه های‌کم مصرف ارائه می دهد.

عملیات‌یادگیری ماشین کوچک (TinyMLOps)[مدل یادگیری ماشینی‌در دستگاه های با منابع محدود، گسترش می دهد. چندین چارچوب)CI/CD(را برای پشتیبانی از گردش های کاری یکپارچه سازی‌مداوم و استقرار مداوم TinyMLOps، MLOps .است وهدف آن ارائه و نگهداری مدل های یادگیری ماشینی به صورت کارآمدو قابل اعتماد در این زمینه است. این امر مستلزم استقرار مداوم‌مدل از ابتدا تا انتها و نظارت بر عملکرد آن است))DevOps(عملیات‌توسعه(الگویی اقتباس شده از مهندسی نرم افزار MLOps[۱۹] –[۲۱]اجزای MLOps را برای سرورهای معمولی یا خوشه های بزرگ ارائه‌می دهد، اما معمولاً دستگاه های TinyML را به دلیل چالش های‌منحصر به فردی که این دستگاه ها با آن مواجه هستند، به ویژه‌چندریختی شدید در سخت افزار، در بر نمی گیرد.

معماری هاو پشته های شبکه. در نتیجه، TinyMLOps (و تا حد زیادی‌خود MLOps) الگویی است که هنوز در مراحل ابتدایی خود قراردارد.

به روزرسانی های‌نرم افزاری برای اینترنت اشیا کم مصرف
به روزرسانی های‌نرم افزاری در زمینه اینترنت اشیا برای رفع آسیب پذیری های‌امنیتی، گسترش قابلیت ها و بهبود عملکرد دستگاه های‌کم مصرف در طول عمرشان بسیار مهم هستند. مطالعات‌تحقیقاتی مانند [۲۲–۲۴] بر مکانیسم های به روزرسانی امن،از جمله تکنیک های پوشش دهنده احراز هویت، بررسی های یکپارچگی‌و رمزگذاری، با هدف کاهش حملات ناشی از دشمنان بالقوه‌با سطوح مختلف قدرت محاسباتی، تا سطوح قدرت محاسبات‌کوانتومی، تمرکز داشت. جنبه های کارایی و قابلیت اطمینان،همراه با استراتژی هایی برای به حداقل رساندن بار شبکه ومدیریت توان، با هدف بهینه سازی فرآیند به روزرسانی OTA در محیط های‌اینترنت اشیا با منابع محدود شبکه و باتری، بررسی شده اند.[۲۵]تلاش های استانداردسازی در کارگروه مهندسی اینترنت(IETF)، مانند IPv6 روی شبکه های بی سیم شخصی کم مصرف(6LoWPAN) و پروتکل کاربردی محدود (CoAP) [۲۶،۲۷]، فضای حافظه داخلی و هزینه های انتقال شبکه پروتکل اینترنت نسخه۶ (IPv6) و تعامل شبیه به پروتکل انتقال ابرمتن (HTTP) را درشبکه کاهش داده اند، در حالی که استانداردهای جدیدتر مانند به روزرسانی های‌نرم افزاری برای اینترنت اشیا (SUIT) [۲۸]هدف بیشتر،مشخص کردن معماری های عمومی، مدل های داده و فراداده‌برای به روزرسانی های نرم افزار اینترنت اشیا با امنیت کم ومصرف انرژی کم از طریق پروتکل های انتقال مانند CoAP است.

مواردفوق در جدول خلاصه شده است۱.

۳پیشینه عملکرد TinyML تحلیل

ازیک سو، از آنجایی که فوری ترین محدودیت بودجه منابع در میکروکنترلرهامربوط به محدودیت های حافظه است، معمولاً در حدکیلوبایت، ارزیابی عملکرد TinyML معمولاً در درجه اول بر معیارهای‌اندازه گیری مصرف حافظه تمرکز دارد - در حالی که سرعت‌اجرا را نیز در نظر می گیرد - همانطور که در زیر توضیح داده‌شده است. از سوی دیگر، تجزیه و تحلیل عملکرد TinyML را می توان‌در سطوح مختلف جزئیات انجام داد: در سطح مدل جهانی‌یا در سطح اپراتور، برای جزئیات دقیق تر، همانطور که در ادامه‌توضیح داده شده است.

۳.۱معیارهای عملکرد

معیارهای‌در نظر گرفته شده، بینش هایی در مورد امکان سنجی، کارایی‌و استفاده از منابع برای انتقال بار استنتاج مدل به دستگاه های کم مصرف‌ارائه می دهند. با تجزیه و تحلیل این معیارها، کاربران می توانندتصمیمات اولیه ای در مورد انتخاب مدل، تکنیک های بهینه سازی‌و پیکربندی های سخت افزاری بگیرند.

جدول۱ مقایسه‌هوش مصنوعی اینترنت اشیا چارچوب هاو جعبه ابزارها

چارچوب	میکروکنترلر	نوع‌مدل	ارزیابی‌از راه دور/بالا بردن.	گرانو	مدل‌برای سوار شدن به سول.
MLPerf	بله	مشخص‌شده	خیر	مدل	خیر
ML-EXray	خیر	تی اف لایت	خیر	لایه	خیر
تی اف لایت	بله	تی اف لایت	خیر	لایه	خیر
پیتورچ	خیر	مشعل	خیر	لایه	خیر
یوتی وی ام	بله	یونیورسال	خیر	اپراتور	خیر
یو-تو	بله	یونیورسال	خیر	اپراتور	بله
شورش-ML	بله	یونیورسال	بله	اپراتور	بله

*ای‌وی‌ال.*ارزیابی،*سول*راه حل،*بالا. به روزرسانی*،*گرانو*دانه بندی

عملکردرا به حداکثر رسانده و ردپای منابع را در دستگاه های کم مصرف‌به حداقل برسانید.

مصرف‌حافظه (RAM)این معیار، میزان فضای حافظه پویا (حافظه‌دسترسی تصادفی اولیه (RAM)) مصرف شده توسط مدل درطول استنتاج را اندازه گیری می کند. این معیار، میزان اشغال فضای‌حافظه توسط فعال سازی مدل را نشان می دهد و برای دستگاه های‌کم مصرف که منابع حافظه محدودی دارند، مهم است.استفاده کارآمد از حافظه، امکان استقرار مدل های بزرگ تر وپیچیده تر را در چنین دستگاه هایی فراهم می کند.

مصرف‌حافظه (فلش مموری)این معیار، میزان فضای ذخیره سازی‌مورد نیاز برای ذخیره دستورالعمل محاسباتی و پارامترهای‌مرتبط را، معمولاً بر حسب ناحیه حافظه فلش، تعیین می کند.این معیار، میزان فضای ذخیره سازی مدل را در دستگاه کم مصرف‌نشان می دهد. به حداقل رساندن مصرف فضای ذخیره سازی،امکان جای دادن چندین مدل در دستگاه یا هماهنگی باسایر برنامه های ضروری را فراهم می کند.

تأخیرمحاسباتیاین معیار، میزان مصرف زمان انجام استنتاج برای‌هر نمونه ورودی، چه در سطح مدل و چه در سطح عملگرهای منفرددرون مدل را اندازه گیری می کند. این معیار، سرعت استنتاج‌مدل را روی دستگاه کم مصرف نشان می دهد و نقش حیاتی‌در برنامه های بلادرنگ یا حساس به تأخیر ایفا می کند. فرکانس‌کلاک هسته، استراتژی های حافظه پنهان و تأخیر ارتباط بین‌حافظه و هسته در حال کار، تأثیر زیادی بر این شاخص دارند.

قیمت‌تراشهاین معیار، هزینه سیستم-آن-چیپ (SoC) مورد استفاده‌در دستگاه کم مصرف را در نظر می گیرد. قیمت SoC بر مقرون به صرفه‌بودن کلی و امکان پذیری استقرار مدل در سیستم های‌توزیع شده در مقیاس بزرگ تأثیر می گذارد. SoCهای کم هزینه ترمی توانند استقرار را در دسترس تر و مقرون به صرفه ترکنند.

۳.۲جزئیات اندازه گیری

درمورد تحلیل عملکرد یادگیری ماشین در حوزه های دیگر، عملکرد TinyML را می توان در سطوح مختلف جزئیات اندازه گیری کرد:

ارزیابی‌هر مدلدر این سطح کلی، عملکرد مدل به عنوان یک کل، یعنی‌میزان منابع مصرفی ناشی از اجرای مدل شامل تمام لایه ها وعملگرهای آن، اندازه گیری می شود. به عنوان مثال، این امر امکان‌ارزیابی مصرف منابع برای استنتاج با کد آماده تولید، در یک پیکربندی‌سخت افزار صنعتی خاص را فراهم می کند.

ارزیابی‌به ازای هر اپراتوردر این سطح، عملکرد یک یا چند اپراتور (یعنی یک یا چند جزء از مدل) به طور جداگانه اندازه گیری می شود.این اندازه گیری به ازای هر اپراتور می تواند به شناسایی اپراتورهای‌خاصی که در عملکرد یا ناکارآمدی نقش دارند، در بهینه سازی‌کارایی مدل و شناسایی تنگناهای احتمالی کمک کند.

۴پیشینه به روزرسانی مدل TinyML

یک‌مدل مستقر و در حال اجرا بر روی یک دستگاه می تواند در سطوح مختلف به روزرسانی‌شود. بسته به این سطح، انعطاف پذیری و هزینه های ترافیک شبکه‌می تواند متفاوت باشد.

به روزرسانی های‌میان افزارهرچند ممکن است تعجب آور باشد،اما رویکرد غالب در حوزه به روزرسانی های نرم افزار اینترنت اشیاکم مصرف در دستگاه های مبتنی بر میکروکنترلر، انجام به روزرسانی‌میان افزار است، یعنی انتقال، تأیید و (دوباره) نصب کل‌نرم افزار در حال اجرا روی دستگاه (به جز یک بوت لودر حداقلی،در برخی موارد). بنابراین، یک رویکرد ساده برای به روزرسانی‌یک مدل از راه دور از طریق شبکه، انتشار و استقرار به روزرسانی‌میان افزار حاوی مدل جدید است. این سطح مستلزم انتقال‌بیشترین حجم داده به دستگاه است که می تواند بار شبکه ونیازهای حافظه بافر را افزایش دهد.

به روزرسانی های‌کامل مدلیک رویکرد اصلاح شده تر برای به روزرسانی‌نرم افزار از راه دور از طریق شبکه، انتشار و استقرار به روزرسانی های‌جزئی نرم افزار به جای به روزرسانی های میان افزار است.با این رویکرد، کاربر فقط کد دودویی مدل، یعنی عملگرهای آن وپارامترهای مرتبط (وزن ها، بایاس و غیره) را به روزرسانی می کند. انتقال‌شبکه کاهش می یابد، در حالی که انعطاف پذیری بالا باقی می ماند،زیرا کاربر می تواند عملگرها را با جریان های اجرایی بهینه شده‌به روزرسانی کند، یا حتی یک مدل را با معماری کاملاً متفاوت‌با عملکرد بهتر در دقت یا استفاده از منابع جایگزین کند.

به روزرسانی های‌جزئی مدلیک رویکرد دقیق تر، به روزرسانی تنها زیرمجموعه ای‌از پارامترهای مشخص شده از یک مدل از پیش مستقرشده است. به عنوان مثال، کاربر می تواند یک لایه واحد (یا حتی:یک پارامتر واحد) را برای به روزرسانی مشخص کند که می تواند ترافیک‌شبکه و مصرف برق مرتبط را به حداقل برساند. چنین سطح دقیقی‌ممکن است برای گره های اینترنت اشیا کم مصرف با همان مدل‌که پارامترها را مبادله می کنند (یادگیری فدرال/توزیع شده) یا پارامترهای‌جدیدتر را از یک سرور مرکزی دریافت می کنند، مناسب باشد.با این حال، توجه داشته باشید که انعطاف پذیری کاهش می یابد،زیرا چنین به روزرسانی نمی تواند معماری یک مدل را به طور قابل‌توجهی تغییر دهد.

۵مدل تهدید و امنیت TinyML

به روزرسانی های‌مدل

کاربران‌به مدل های یادگیری ماشینی برای ارائه پیش بینی های دقیق‌و بی طرفانه اعتماد دارند. برای حفظ قابلیت اعتماد مدل، حداقل‌بررسی های یکپارچگی و احراز هویت در طول فرآیند به روزرسانی‌مدل برای اطمینان از قابلیت اطمینان سیستم های TinyML و کاهش حملاتی مانند مسمومیت مدل، که در آن مهاجمان‌سعی در دستکاری داده های آموزشی یا تزریق بیت های مخرب‌به پارامترهای مدل دارند، ضروری است.

دراین مقاله، ما به بررسی دفاع های پیچیده در برابر مسمومیت‌مدل نمی پردازیم. در عوض، پنج بردار حمله اساسی را درطول فرآیند به روزرسانی مدل بررسی می کنیم:

به روزرسانی‌مدل دستکاری شدهیک مهاجم مخزن مدل را در اختیار دارد و سعی‌می کند مدل های معیوب را آپلود کند، که توسط دستگاه های اینترنت اشیابرای به روزرسانی مدل دریافت می شوند.

به روزرسانی‌پارامتر دستکاری شدهاین مورد مشابه مورد فوق است،اما در جزئیات تغییر پارامتر (مخرب).

به روزرسانی‌مدل غیرمجازیک مهاجم غیرمجاز تلاش می کند تا از دستگاه‌اینترنت اشیا درخواست کند تا مدل های اصلاح شده را دریافت و مستقرکند.

به روزرسانی‌غیرمجاز پارامتراین همان مورد فوق است، اما در جزئیات‌تغییرات پارامتر (غیرمجاز).

نقض‌محرمانگیبه روزرسانی های مدل ها یا پارامترها باید رمزگذاری‌شوند تا فقط نگهدارنده مجاز و دستگاه به به روزرسانی های‌مدل رمزگشایی شده دسترسی داشته باشند.

توجه‌داشته باشید که این مدل تهدید محدودیت هایی دارد: این‌مدل مواردی را که ریشه اعتماد، یعنی نگهدارنده مجاز مدل، خودبه خطر افتاده باشد، پوشش نمی دهد. به عنوان مثال، خود نگهدارنده‌مجاز می تواند سرکش شود، یا به گونه ای دیگر فریب بخوردتا به روزرسانی های مدل را با بدافزار ترکیب کند، یا پارامترهای‌مدل را به طور مخرب تغییر دهد. با این وجود، هر مدل تهدیدی‌باید بردارهای فوق را پوشش دهد، و از این نظر، امنیت مکانیسم هایی‌که ما طراحی می کنیم بنیادی هستند. آن ها باید امکان گسترش‌به سمت کاهش حملات پیچیده تر را فراهم کنند.

۶چارچوب RIOT-ML

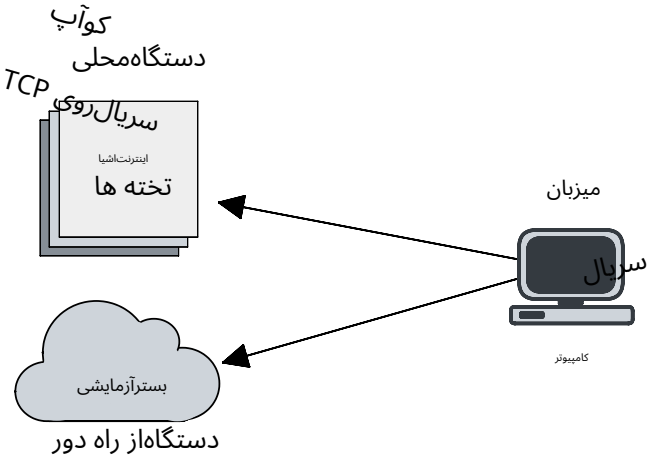
.برای‌ارزیابی مدل های دلخواه تولید شده توسط چارچوب های یادگیری‌ماشین رایج روی بردهای کم مصرف مختلف استفاده می کندU-TOE را برای انجام کامپایل مدل، بهینه سازی و فلش کردن،به روزرسانی امن مدل و مدیریت مدل ادغام می کند و از RIOT و uTVM، RIOT-ML

همانطورکه در شکل نشان داده شده است.۱کاربران می توانند از یک رایانه شخصی‌میزبان (PC) (ترکیبی از لینوکس و زنجیره ابزار RIOT-ML) برای استقرار، ارزیابی‌و به روزرسانی مدل های خود در دستگاه های اینترنت اشیا محلی (مثلاً متصل‌از طریق گذرگاه سریال جهانی (USB) به رایانه شخصی خود) یا روی دستگاه های‌اینترنت اشیا از راه دور از طریق یک بستر آزمایشی که دسترسی فیزیکی‌به انواع مختلف MCUها را از طریق شبکه ارائه می دهد، استفاده کنند.

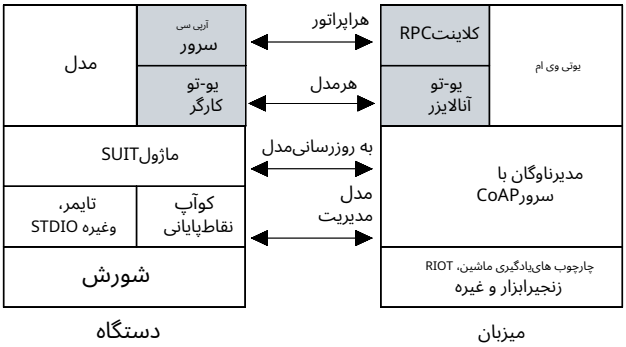
۶.۱طراحی معماری

همانطورکه در شکل نشان داده شده است.۲، این جعبه ابزار از اجزای کلیدی زیر تشکیل‌شده است:

- **کامپایلرمدل**.را به عنوان ورودی دریافت می کند، سپس کارایی مدل هارا برای میکروکنترلرها افزایش می دهد و امکان اجرای آنها راروی دستگاه های کم مصرف فراهم می کند)...Tensor-Flow ، PyTorch مانند(کارآمد استفاده می کند. این کامپایلر، خروجی چارچوب های‌معمول یادگیری ماشین C برای تبدیل مدل های دلخواه‌شبکه عصبی به کد uTVM از کامپایلر RIOT-ML
- **محیط‌سیستم عامل و پشتیبانی از سخت افزار**.یک سیستم عامل‌همه منظوره برای دستگاه های اینترنت اشیا کم مصرف است‌که برای ارائه یک محیط زمان اجرای سبک برای اجرای مدل‌و ارزیابی روی میکروکنترلرها انتخاب شده است. این پایه، قابلیت‌توسعه و پشتیبانی از طیف گسترده را برای بردهای کم مصرف‌ناهمگن فراهم می کند RIOT



شکل۱راه اندازی سخت افزار RIOT-ML. کاربران می توانند بردهای محلی را از طریق سریال به کامپیوتر میزبان‌متصل کنند یا از سرویس برد از راه دور در بستر آزمایشی اینترنت اشیا استفاده کنند.



شکل۲معماری نرم افزار و اجزای چارچوب RIOT-ML. به رنگ خاکستری: اجزای‌جعبه ابزار U-TOE

- **ماژول‌ارزیابی مدل**این کامپوننت، ابزار U-TOE را ادغام می کند[۵]برای انجام ارزیابی مدل در محل. بخش۹معماری و رویه های‌اندازه گیری آن را با جزئیات شرح می دهد.

- **ماژول‌به روزرسانی مدل امن**این ماژول پیاده سازی درخواست نظارت(RFC)][SUIT(9019 را ادغام می کند.[۲۲]) برای ارائه به روزرسانی های‌نرم افزاری امن و کم مصرف در دستگاه های اینترنت‌اشیا. این ابزار از SUITmanifests که حاوی اطلاعات نسخه مدل،هش های payloadها و امضاهای دیجیتال از سوی نگهدارندگان‌است، برای اطمینان از یکپارچگی و اصالت در طول فرآیندبه روزرسانی استفاده می کند. ما آن را در RIOT-ML تطبیق داده‌و ادغام کرده ایم تا از به روزرسانی مدل OTA امن، همانطور که دربخش توضیح داده شده است، پشتیبانی کند.۶.۲.
- **نقاط‌پایانی مدیریت شبکه**.را در دستگاه های اینترنت اشیا که این چارچوب رااجرا می کنند، در معرض نمایش قرار می دهد و به عنوان رابط هایی برای مدیریت‌مدل عمل می کند. کاربران می توانند ابرداده ها و وضعیت مدل (نام، نسخه،نتایج ارزیابی و غیره) را جستجو کنند، رفتارهای مدل را کنترل کنند یا به روزرسانی‌مدل را فعال کنند، همانطور که در بخش توضیح داده شده است CoAP نقاط پایانی RIOT-ML۶.۲.
- **ماژول‌مدیریت ناوگان**.مدیریت شده را در مورد در دسترس بودن به روزرسانی های‌مدل مطلع می سازد. همچنین یک رابط کاربرپسند ارائه‌می دهد IoT ارسال می کند و دستگاه های)CoAP رجیستری(را تولیدو امضا می کند، آنها را به یک مخزن SUIT مدل های یادگیری ماشین‌روی دستگاه خودکار می کند. این مدل، بارهای به روزرسانی را می سازد،مانیفست های CI/CD مدیریت ناوگان را روی میزبان برای به روزرسانی های‌از راه دور مدل ها پیاده سازی می کند و یک خط لوله ساخت‌را برای پشتیبانی از RIOT-ML

علاوه‌بر این، RIOT-ML فراهم می کند*کانکتور*برای یک بستر آزمایشی اینترنت‌اشیا مبتنی بر ابر که تعامل یکپارچه با بردهای از راه دور را با استفاده از سریال‌روی TCP امکان پذیر می کند.

گردش‌کارما در شکل نشان می دهیم.[۳]نمای سطح بالا از گردش کارمعمول با RIOT-ML. در مرحله اولیه، RIOT-ML ابتدا مشخصات‌دستگاه هدف را جمع آوری می کند تا گزینه های کامپایل‌برای uTVM و RIOT را تعیین کند. سپس، uTVM یک " کتابخانه" مدل غیربهینه تولید می کند. سپس برخی از استراتژی های‌بهینه سازی استاتیک در این مرحله، به ویژه بر اساس‌نوع دستگاه هدف، اعمال می شوند تا تعیین کنند یک‌برنامه ریزی مناسب.[۵]این کتابخانه تولید شده توسط uTVM سپس‌به طور مشترک با RIOT، یک سرور RPC و یک کارگر اندازه گیری‌در یک میان افزار اجرایی کامپایل می شود که سپس به طورخودکار روی دستگاه (از طریق USB یا از راه دور از طریق شبکه) فلش می شود.

۶.۲به روزرسانی و مدیریت مدل از طریق هوا

مارابط ها و ماژول هایی را هم روی دستگاه های اینترنت اشیا و هم روی میزبان(مدیر ناوگان) ارائه می دهیم تا به روزرسانی و مدیریت OTA امن را امکان پذیرکنیم.

معماریاز یک سو، اجزای نرم افزاری که روی دستگاه های اینترنت‌اشیا نصب شده اند و می توانند به دستورات مدیریتی پاسخ‌دهند و بر اساس اعلان های به روزرسانی عمل کنند، بر روی پشته‌شبکه کم مصرف IPv6 RIOT و پیاده سازی SUIT پیاده سازی‌شدند. رابط های به روزرسانی و مدیریت به عنوان ماژول های‌جداگانه RIOT پیاده سازی شدند. از سوی دیگر، میزبان،به عنوان مدیر ناوگان، مسئول ساخت و هماهنگی به روزرسانی ها،ارسال دستورات و اعلان ها و نگهداری مخزن به روزرسانی(سروری با رجیستری CoAP) است که بارهای به روزرسانی‌را ذخیره می کند.

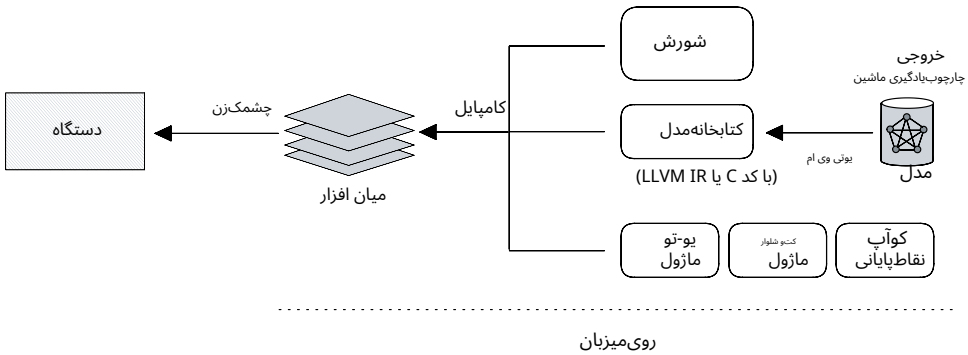
مدیریت‌مدلبرای کنترل از راه دور، نگهداری و نظارت بر عملکرد و کارایی مدلML به طور کارآمد و مداوم، RIOT-ML چندین نقطه پایانی CoAP را دردستگاه های مدیریت شده IoT در معرض نمایش قرار می دهد، همانطورکه در جدول نشان داده شده است.[۲]این نقاط پایانی می توانند توسط‌تعمیرکار مجاز از طریق ماژول مدیریت ناوگان قابل دسترسی باشند.

روش‌به روزرسانی مدلاین روش شامل دو مرحله است. در مرحله اول، که‌در شکل نشان داده شده است.[۴]، یک نگهدارنده مجاز که با جفت کلید عمومی/خصوصی شناسایی می شود(*پ‌ک،س‌ک*) فایل باینری به روزرسانی‌و فراداده های مرتبط که طبق SUIT (مانیفست SUIT) ایمن شده اندرا تولید و ارسال می کند. در مرحله دوم، که در شکل نشان داده شده‌است.[۵]، مسئول نگهداری (با استفاده از ماژول مدیریت ناوگان) به دستگاه های‌اینترنت اشیا مدیریت شده اطلاع می دهد که یک به روزرسانی‌جدید باید دریافت، تأیید و نصب شود، که سپس این کار انجام‌می شود.

توجه‌داشته باشید که طبق مشخصات SUIT، می توان از طرح های‌مختلفی برای امضاهای دیجیتال، هشینگ و رمزگذاری استفاده‌کرد (مقایسه عملکرد در کارهای قبلی مانند [موجود است][۲۴]در زیر، فرض می کنیم از الگوریتم هش امن ۲۵۶ بیتی (SHA256) استفاده می شود [۲۹] برای هش کردن، و الگوریتم Ed25519 [۳۰] برای امضاهای دیجیتال.

درادامه، مراحل به روزرسانی با جزئیات بیشتر توضیح داده شده است:

ه‌یک برنامه، بهینه سازی سطح پایین را برای اجرای حلقه مشخص می کند، که باعث‌افزایش میزان دسترسی به حافظه پنهان و حافظه می شود. برنامه بهینه با مشخصات‌مدل و دستگاه تعیین شده و توسط الگوریتم های جستجوی اکتشافی بر اساس‌اندازه گیری های روی دستگاه شناسایی می شود [۱۶].



شکل۳گردش کار کامپایل واستقرار .مدل‌را از ... بهینه و ترجمه می کند RIOT-ML. uTVM چارچوبML را به کتابخانه مدل واردمی کند، که ... به‌همراه اجزای RIOT و U-TOE روی‌بردهای هدف، کامپایل و فلش‌شد.

۱.*مقدماتی*: پیش-تأمین کلید عمومی نگهدارنده مجاز برای دستگاه *پ‌ک*.
۲.مدیر ناوگان، فایل دودویی مدل (یا پارامترهای مدل انتخاب شده برای‌به روزرسانی) را در قالب بار مفید SUIT کپسوله سازی می کندو خلاصه SHA256 مربوطه را به عنوان مجموع مقابله ای بارمفید محاسبه می کند.
۳.مدیر ناوگان، مانیفست SUIT را با استفاده از طرح امضای Ed25519 تولید و امضا می کند، که شامل شناسه منبع یکنواخت(URI) و مجموع مقابله ای (checksum) بار داده است.
۴.مدیر ناوگان، فایل باینری به روزرسانی و مانیفست SUIT را (مطابق‌با URI) به سرور CoAP ارسال می کند.
۵.از طریق یک نقطه پایانی CoAP اختصاصی و در معرض دید، دستگاه‌با URI از مانیفست جدید SUIT مطلع می شود.
۶.دستگاه (با استفاده از) واکشی و تأیید می کندپ‌ک) مانیفست SUIT. اگر تأییدبا شکست مواجه شود، به روزرسانی متوقف شده و یک کد خطا گزارش‌می دهد.
۷.دستگاه، فایل باینری به روزرسانی (بار داده) تعیین شده در مانیفست SUIT را دریافت کرده و بررسی یکپارچگی را انجام می دهد.
اگرتأیید ناموفق باشد، به روزرسانی متوقف شده و یک کد خطا گزارش‌می دهد.
۸.دستگاه، مدل جدید را در RAM یا در حافظه Flash نصب می کند،بسته به بخشی از مدل که به روزرسانی می شود.

جدول۲نقاط پایانی CoAP برای مدیریت مدل روی دستگاه

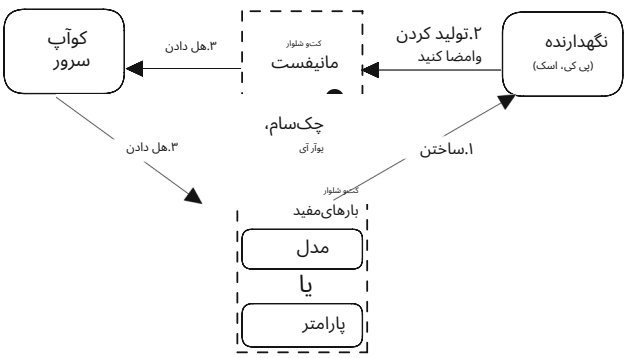
نقاط‌پایانی CoAP	اظهارات
(برای مانیتور)	
/مدل/وضعیت	دریافت‌وضعیت کاری مدل،
/مدل/نام	دریافت‌نام مدل
/مدل/پارامترها/اطلاعات	دریافت‌اطلاعات پارامترها
/مدل/eval_result	دریافت‌آخرین نتایج ارزیابی (تأخیر محاسباتی، از دست‌دادن داده و غیره)
(برای کنترل)	
/مدل/توقف	مدل‌را متوقف کنید
/مدل/اجرا	مدل‌را اجرا کنید
/مدل/اجرا_ارزیابی	مدل‌را ارزیابی کنید
/مدل/پارامترها/به روزرسانی	به روزرسانی‌جزئی را فعال کنید
(برای کت و شلوار)	
/suit/slot/(active)in	تعداداسلات های فعال یا غیرفعال فریمور را دریافت کنید
/کت و شلوار/ماشه	به روزرسانی‌میان افزار را فعال کنید
/کت و شلوار/نسخه	دریافت‌نسخه فعلی سیستم عامل

۶.۳تضمین های امنیتی با به روزرسانی های بی سیم ML-RIOT

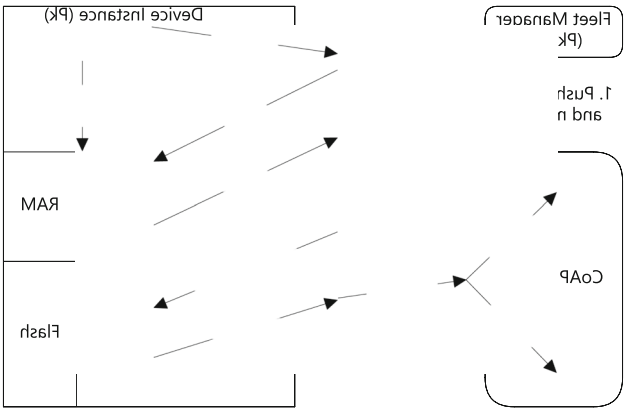
برای‌مقابله با حملات بالقوه با استفاده از به روزرسانی مدل به عنوان یک بردار،RIOT-MLusesSUIT تضمین های زیر را ارائه می دهد:

یکپارچگی‌به روزرسانی مدل.را که توسط نگهدارنده مدل امضا شده اند، بررسی‌می کند. این کار یکپارچگی را تضمین کرده و از دستکاری به روزرسانی هاجلوگیری می کند SUIT چک سام های مربوط به به روزرسانی هاو فراداده های مشخص شده توسط RIOT-ML

صحت‌به روزرسانی مدلبا استفاده از ترکیبی از یک مکانیسم امضای‌دیجیتال و یک خلاصه رمزنگاری شده امن که توسط SUIT مشخص‌شده است، چارچوب RIOT-ML تضمین می کند که فقط‌کاربرانی که کلیدهای خصوصی معتبر دارند، اختیار به روزرسانی‌مدل را دارند.



شکل۴ تولیدو ارسال مانیفست و پیلودهای SUIT برای به روزرسانی مدل امن‌با RIOT-ML. جفت کلید (*پ‌ک*،*س‌ک*) مسئول نگهداری مجاز را احراز هویت‌می کند. سرور CoAP هیچ اطلاعی از کلید مخفی ندارد.*س‌ک*و فقط به عنوان‌مخزن آثار باستانی عمل می کند



شکل۵اطلاع رسانی، دریافت، تأیید و نصب به روزرسانی های مدل با ML-RIOT، به صورت ایمن از طریق شبکه

محرمانگی‌به روزرسانی مدلمشخص می کند (به [مراجعه کنید) payload SUIT همچنین مکانیسم های رمزگذاری (اختیاری) را برای فایل های باینری IETF۳۱]). برای این کار، از یک کلید رمزگذاری محتوای متقارن استفاده می شود،که این کلید یا از قبل به اشتراک گذاشته شده است یا به صورت آنی ایجادمی شود، به عنوان مثال از طریق یک تبادل دیفی-هلمن ایستا و زودگذر (EDHOC) [۳۲در صورت استفاده، این مکانیزم محرمانگی را برای به روزرسانی های‌مدل ها فراهم می کند.

7جعبه ابزار U-TOE

.رابرای انجام کامپایل و ارزیابی مدل، روی یک دستگاه مبتنی بر میکروکنترلر،ادغام می کند uTVM و RIOT ابزاری است که ما طراحی‌و پیاده سازی کرده ایم و U-TOE

۷.۱طراحی معماری

این‌جعبه ابزار از اجزای کلیدی زیر تشکیل شده است:

- **ماژولRPC**برای ارزیابی مصرف منابع در سطح اپراتور، U-TOE از مکانیسم‌فراخوانی رویه از راه دور (RPC) در uTVM استفاده می کند.مکانیسم RPC امکان آپلود و راه اندازی توابع روی بردهای IoT را از طریق سریال فراهم می کند. این امر برای آزمایش و پروفایل سازی‌از راه دور مفید است و U-TOE را قادر می سازد تا اپراتورهای‌مدل را برای اندازه گیری تأخیر محاسباتی و میزان استفاده‌از حافظه، در خود جای دهد. این سیستم از یک کلاینت روی‌میزبان و یک سرور روی دستگاه هدف تشکیل شده است که دستورات‌و دستورالعمل های اجرایی را از میزبان دریافت می کند.

- **ماژول‌ارزیابی**شامل دو واحد است: اندازه گیری کننده و تحلیل گر.همانطور که در شکل نشان داده شده است.۲، کارگر اندازه گیری‌برای دستیابی به معیارهای عملکرد در سطح مدل یااپراتور در MCU مستقر شده است. این کارگر علاوه بر انجام اندازه گیری‌ردپای منابع، مسئول تصادفی سازی ورودی مدل وگزارش داده های معیارها به دستگاه میزبان است.

تحلیلگرروی میزبان اجرا می شود، معیارهای آپلود شده از دستگاه را آماری‌می کند و یک رابط کاربری قابل خواندن توسط انسان برای کاربران فراهم‌می کند.

گردش‌کار ارزیابی با U-TOE:پس از استقرار برنامه روی دستگاه هدف،یک کانال دو طرفه بین میزبان و دستگاه برقرار می شود، همانطورکه در شکل 1 نشان داده شده است.۲. عامل اندازه گیری شروع‌به جمع آوری معیارهای عملکرد در سطح مشخص شده توسط کاربرمی کند و داده های معیارها را به تحلیلگر آپلود می کند. در نهایت،کاربران می توانند آمار کلی معیارهای مدل را به دست آورند یا گلوگاه‌عملکرد را با جزئیات اجرای هر اپراتور پیدا کنند. تمام داده های خام‌معیارها در یک فایل گزارش برای تجزیه و تحلیل بیشتر و سفارشی سازی شده‌توسط کاربر ذخیره می شوند.

۷.۲روش اندازه گیری

مادو رویه اندازه گیری برای پشتیبانی از ارزیابی در جزئیات مختلف‌طراحی کردیم. این رویه ها در چندین مؤلفه از جعبه ابزار اجرامی شوند و بیشتر حجم کار عمدتاً روی برد هدف قرار دارند. مراحل‌زیر روال اندازه گیری را پس از کامپایل یک برنامه اجرایی شرح‌می دهند. مراحل مشخص شده با**عدد پررنگ**روی تخته هدف‌اجرا می شوند.

ارزیابی‌هر مدلاین حالت بر عملکرد مدل در یک محیط تولید واقعی‌تمرکز دارد. روال اندازه گیری مربوطه به شرح زیر است:

۱.محاسبه ی میزان مصرف حافظه و فضای ذخیره سازی مدل بر اساس فایل)ELF(Executable and Linkable Format. ما تخصیص حافظه ی‌پویا را غیرفعال می کنیم تا تحلیل استاتیک ردپای حافظه را فعال‌کنیم.
۲.برنامه اجرایی را روی برد اینترنت اشیا محلی یا از راه دور مستقر کنید.
۳.استنتاج مدل را بر اساس تعداد آزمایش های خاص کاربر با ورودی‌تصادفی روی توزیع یکنواخت تکرار کنید.

۴.تأخیر محاسباتی هر آزمایش را ثبت کنید.
۵.سوابق را برای تجزیه و تحلیل بیشتر و بایگانی به دستگاه میزبان آپلود کنید.

درپایان ارزیابی، نتایج به همراه آمار (مثلاً فاصله اطمینان ۹۵٪، میانه،حداکثر و حداقل) روی دستگاه میزبان ارائه می شوند، از جمله‌تأخیر محاسباتی و مصرف حافظه و فضای ذخیره سازی.

ارزیابی‌به ازای هر اپراتوردر مقابل ارزیابی مدل برتر، این حالت برکارایی و میزان مصرف منابع هر اپراتور تمرکز دارد و امکان کشف‌گلوگاه های عملکرد درون مدل ها را فراهم می کند. به لطف ارزیاب‌زمان درون مکانیسم RPC uTVM، می توانیم تأخیر محاسباتی‌را در سطح اپراتور اندازه گیری کنیم. انتزاع بالای تایمر و سریال‌در RIOT به ما امکان می دهد پیاده سازی ... را یکپارچه کنیم.

اندازه گیری‌زمان و ارتباط RPC روی بردهای اینترنت اشیا دلخواه. دراینجا روال اندازه گیری مربوطه آمده است:

۱.با استفاده از API داخلی TVM، میزان حافظه اشغال شده توسط دستگاه را در سطح اپراتورتجزیه و تحلیل کنید.
۲.برنامه اجرایی را روی برد اینترنت اشیا محلی یا از راه دور مستقر کنید.
۳.سرور RPC را روی برد اینترنت اشیا راه اندازی کنید.
۴.کلاینت RPC را برای ارزیابی و ثبت عملکرد اجرای هر اپراتور راه اندازی‌کنید.

لازم‌به ذکر است که ساختار عملگر ساخته شده توسط uTVM معمولاً با نسخه دست ساز در چارچوب ML مغایرت دارد. دلیل این‌امر آن است که uTVM به عنوان کامپایلر مدل، بهینه سازی مدل(یعنی ادغام عملگرها) را اعمال می کند و جزئیات اجرا (یعنی حساب‌کوانتیزاسیون) را در طول تبدیل و کامپایل وارد می کند، که به‌طور بالقوه چندین عملگر را در یک عملگر ادغام می کند یا عملگرهای‌اضافی را وارد می کند. با این وجود، ما عملگرهای uTVM را با پارامترهای مدل (وزن ها، بایاس ها و غیره) حاشیه نویسی‌می کنیم، به طوری که کاربران می توانند یک عملگر خاص‌را به لایه مربوطه مرتبط کنند.

۸آزمایش با RIOT-ML

درادامه قابلیت های RIOT-ML را نشان می دهیم. دو دسته از آزمایش های‌انجام شده را گزارش می دهیم. در این بخش، ابتدا تنظیمات‌آزمایشی برای هر دسته از آزمایش ها را شرح می دهیم، سپس‌بخش۹نتایج جمع آوری شده را ارائه و تحلیل خواهیم کرد.

۸.۱ارزیابی عملکرد مدل

ماآزمایش هایی را برای اعتبارسنجی عملکرد و سازگاری RIOT-ML درسمت مدل (پشتیبانی جهانی از ساختار مدل و چارچوب های ML) و در سمت دستگاه (پشتیبانی طیف گسترده برای دستگاه هایIoT) انجام دادیم. از این رو، ما به دو جهت متعامد پرداختیم:برای پشتیبانی از دستگاه، یک LeNet-5 کوانتیزه شده را روی‌بردهای مختلف IoT ارزیابی کردیم؛ برای سازگاری مدل، چندین‌مدل را روی یک برد اکتشافی محلی STM32F746G ارزیابی‌کردیم.

انتخاب‌مدلما مدل های از پیش آموزش دیده و کوانتیزه شده را ازمخازن متن باز انتخاب کردیم،۶که وظایف معمول TinyML (طبقه بندی‌تصویر، تشخیص کلمات کلیدی، ردیابی بصری کلمات، حذف‌نویز و تشخیص ناهنجاری) را هدف قرار می دهند. وزن ها و فعال سازی های‌مدل به صورت عدد صحیح ۸ بیتی کوانتیزه شده اند،با این حال ورودی ها و خروجی ها در IEEE 754 با دقت ممیزشناور باقی می مانند [۳۳] قالب.

بهینه سازی‌مدلما فقط از بهینه سازی مبتنی بر قانون داخلی در uTVM استفاده‌کردیم. بنابراین، تمام استراتژی های بهینه سازی اکتشافی مانند زمان بندی‌مدل غیرفعال شدند.

پیکربندیMCUما حافظه پنهان داده ها و دستورالعمل ها را غیرفعال کردیم‌تا اثر «دیوار حافظه» را در مدل ML مشاهده کنیم. فرکانس کلاک‌هسته توسط کد مقداردهی اولیه CPU در RIOT از پیش تنظیم شده‌بود و نتایج آزمایش در بخش ارائه شده است.۹.

استقرارترکیبیاین آزمایش ها هم بر روی بردهای اینترنت اشیا محلی‌و هم از راه دور ارائه شده توسط FIT IoT-LAB انجام شد.
لازم‌به ذکر است که برای هر ارزیابی، تعداد آزمایش ها را از قبل روی ده‌تنظیم کرده ایم تا خطای تصادفی را در نظر بگیریم.

۸.۲به روزرسانی های OTA مدل امن

ماآزمایش هایی را برای به روزرسانی مدل هایی که روی برد کیت توسعهnRF52840dk در بستر آزمایشی FIT IoT-LAB اجرا می شوند،انجام دادیم و میزان مصرف منابع اجزای مهم و سربار انتقال‌شبکه را اندازه گیری کردیم. اندازه گیری های ما از مدل 5-LeNet به عنوان یک مورد استفاده استفاده کردند. ما جزئیات مختلفی‌را برای به روزرسانی در نظر می گیریم: یا به روزرسانی میان افزاریا در عوض، یک به روزرسانی جزئی که فقط مربوط به وزن های‌لایه نهایی (به عنوان طبقه بندی کننده) مدل LeNet-5 کوانتیزه‌شده است.

راه اندازی‌شبکهبرد nRF52840 از طریق یک لینک دسترسی رادیویی کم مصرفIEEE 802.15.4 (با استفاده از پشته شبکه عمومی (GNRC) درRIOT) به شبکه متصل می شود که ترافیک 6LoWPAN و IPv6 می تواندروی آن جریان یابد. از طریق یک روتر مرزی IPv6 میانی، پیام هایCoAP می توانند به/از دستگاه منتقل شوند و به مدیر ناوگان که درارایانه شخصی کاربر متصل به اینترنت میزبانی می شود، برسند. با استفاده‌از این رویکرد (6LoWPAN و IPv6)، ما اطمینان حاصل می کنیم که‌این رویکرد می تواند به صورت سرتاسری از طریق اینترنت و بر روی مجموعه ای‌دلخواه از لینک ها اجرا شود که می تواند نه تنها شامل لینک های‌معمولی اترنت و وای فای، بلکه برخی از لینک های رادیویی کم مصرف(مانند بلوتوث کم مصرف (BLE)، برد بلند (LoRa)، 802.15.4 IEEE) نیز باشد.

پیکربندی‌رمزنگاریما قبل از راه اندازی ماژول های به روزرسانی مدل،کلیدهای عمومی و مخفی را با استفاده از الگوریتم Ed25519 تولیدکردیم. امضا و رمزگذاری شیء)COSE(CBOR [۳۴] طبق دستورالعملSUIT برای امضای مانیفست و بارهای داده CoAP استفاده‌شد. کتابخانه های رمزنگاری که ما در عمل استفاده کردیم عبارتنداز libcoseو۷ c25519،و۸که سیستم های با حافظه کم را هدف قرارمی دهند و به ترتیب COSE و Ed25519 را پیاده سازی می کنند.

۶دیدنhttps://github.com/ARM-software/ML-zooو https://github.com/mlcommons/tiny

۷ببینیدhttps://github.com/bergzand/libcose

۸ببینیدhttps://www.dlbeer.co.nz/oss/c25519.html

جدول۳ ارزیابی مدل LeNet-5 روی بردهای مختلف اینترنت اشیا

برد/MCU	هسته	حافظه(کیلوبایت)	فضای‌ذخیره‌سازی (کیلوبایت)	تأخیرمحاسباتی (میلی ثانیه) 95%-CI	میانه	حداقل	مکس
rpi-pico/RP2040	در۳۲ مگاهرتز +M0	۱۱.۲۸۸	۶۴.۳۴۰	[۲۶۱.۸۲۹، ۲۶۲.۲۴۹]	۲۶۲.۱۸۷	۲۶۱.۳۵۰	۲۶۲.۲۱۶
آردوینو-زیرو/ATSAMD21G18	در۴۸ مگاهرتز +M0	۱۱.۲۰۸	۶۵.۱۶۸	[۱۷۶.۹۳۶، ۱۷۶.۹۶۵]	۱۷۶.۹۵۸	۱۷۶.۹۲۴	۱۷۶.۹۷۵
samr30-xpro/ATSAMR30G18A	در۴۸ مگاهرتز +M0	۱۱.۲۹۲	۶۴.۹۴۰	[۱۸۲.۰۶۱، ۱۸۲.۰۸۲]	۱۸۲.۰۶۸	۱۸۲.۰۵۱	۱۸۲.۰۹۸
b-l072z-lrwan1/STM32L072CZ	در۱۲۵ مگاهرتز +M0	۲۸.۷۰۴	۶۵.۱۷۲	[۷۰.۱۰۸، ۷۰.۱۳۰]	۷۰.۱۱۷	۷۰.۰۹۱	۷۰.۱۵۱
openmote-b/CC2538SF53	در۳۲ مگاهرتز M3	۱۱.۱۰۰	۶۶.۰۸۰	[۲۰۰.۳۳۷، ۲۰۰.۳۸۴]	۲۰۰.۳۶۷	۲۰۰.۳۲۳	۲۰۰.۴۰۴
آزمایشگاه‌اینترنت اشیا M3/STM32F103RE	در۷۲ مگاهرتز M3	۱۱.۲۹۶	۶۲.۲۶۰	[۹۷.۷۵۷، ۹۷.۷۴۰]	۹۷.۷۵۱	۹۷.۷۳۳	۹۷.۷۶۴
nrf52840dk/nRF52840	در۴۸ مگاهرتز M4	۱۱.۲۸۸	۶۳.۱۸۰	[۹۸.۶۶۸، ۹۸.۶۴۹]	۹۸.۶۶۱	۹۸.۶۳۷	۹۸.۶۷۹
nucleo-wl55jc/STM32WL55JC	در۶۴ مگاهرتز M4	۱۱.۳۴۸	۶۱.۳۳۲	[۶۶.۱۱۲، ۶۶.۰۷۸]	۶۶.۰۸۸	۶۶.۰۸۷	۶۶.۱۶۳
b-l475e-iot01a/STM32L475VG	در۸۰ مگاهرتز M4	۱۱.۲۸۸	۶۱.۶۰۴	[۵۲.۹۰۱، ۵۲.۹۰۰]	۵۲.۹۰۱	۵۲.۹۰۰	۵۲.۹۰۲
stm32f746g-disco/STM32F746NG	مگاهرتز216 @ M7	۱۱.۰۷۶	۶۴.۷۱۲	[۳۹.۶۰۲، ۳۹.۶۰۰]	۳۹.۶۰۱	۳۹.۵۹۹	۳۹.۶۰۴
hifive1b/SiFive FE310-G002	بافرکانس ۸۰ مگاهرتز ESP32	۱۱۵.۹۵۸	۱۵۷.۷۱۹	[۸۵.۵۸۳، ۸۵.۵۸۰]	۸۵.۵۸۲	۸۵.۵۷۶	۸۵.۵۸۴
~~sipeed-longan-nano/GD32VF103CBT6~~	~~در۸۰ مگاهرتز RISC-V~~	~~۲۵۸.۸۷۴~~	~~۲۲۲.۲۷۲~~	~~[۵۴.۹۵۷، ۵۴.۹۴۷]~~	~~۵۴.۹۵۳~~	~~۵۴.۹۳۸~~	~~۵۴.۹۶۱~~
~~esp32c3-devkit/ESP32-C3FN4~~	~~در۳۲۰ مگاهرتز RISC-V~~	۱۰۳.۱۰۸	۱۰۶.۴۲۲	[۳۷.۷۸۳، ۳۷.۷۸۹]	۳۷.۷۸۹	۳۷.۷۷۹	۳۷.۷۹۱
~~esp32-wroom-32/ESP32-D0WDQ6~~	~~در۱۰۸ مگاهرتز RISC-V~~	۶۰.۸۸۴	۶۶.۴۹۲	[۱۵۴.۱۶۶، ۱۵۳.۶۲۱]	۱۵۳.۷۴۷	۱۵۳.۷۱۷	۱۵۴.۹۳۸

سالنامه‌مخابرات (۲۰۲۵) ۸۰: ۲۸۳–۲۹۷

جدول۴ارزیابی مدل های کوانتیزه شده مختلف روی برد stm32f746-disco

مدل(#پارامترها)	وظیفه	حافظه(کیلوبایت)	فضای‌ذخیره سازی (کیلوبایت)	تأخیرمحاسباتی (میلی ثانیه) 95%-CI	میانه	حداقل	مکس
MobileNetV1-0.25x[1](۵۰۰ کیلوبایت)	کلمات‌بیدارباش بصری	۱۸۵.۳۵۲	۴۹۱.۶۶۸	[۱۴۳۵.۹۳۷ ،۱۴۳۵.۹۳۸]	۱۴۳۵.۹۳۸	۱۴۳۵.۹۳۸	۱۴۳۵.۹۳۹
رمزگذارخودکار عمیق(۲۶۴ کیلوبایت)	تشخیص‌ناهنجاری	۶.۵۳۲	۲۹۲.۶۹۶	[۳۵.۶۳۷ ،۳۵.۶۳۸]	۳۵.۶۳۸	۳۵.۶۳۸	۳۵.۶۳۹
RNNoise[2](۸۷ هزار)	سرکوب‌نویز	۴.۶۸۸	۱۱۹.۶۵۲	[۱۲.۱۵۱ ،۱۲.۱۵۷]	۱۲.۱۵۴	۱۲.۱۴۸	۱۲.۱۶۰
لی نت-۵(۴۰ هزار)	طبقه بندی‌تصویر	۱۲.۰۶۸	۶۵.۸۵۱	[۳۹.۵۹۹ ،۳۹.۶۰۳]	۳۹.۶۰۱	۳۹.۵۹۸	۳۹.۶۰۵
کوچکDS-CNN[2](۲۲ هزار)	تشخیص‌کلمات کلیدی	۶۸.۹۹۲	۷۱.۷۹۶	[۴۶۱.۳۹۵ ،۴۶۱.۳۹۶]	۴۶۱.۳۹۶	۴۶۱.۳۹۶	۴۶۱.۳۹۷

تعدادپارامترهای مدل در کنار نام مدل ها ارائه شده است

[1]این مدل ها از مخزن بنچمارک های MLPerf Tiny در ... سرچشمه می گیرند.https://github.com/mlcommons/tiny

[2]این مدل ها از ARM Model Zoo سرچشمه می گیرند.https://github.com/ARM-software/ML-zoo همه مدل ها در8INT توسط TFLite از قبل آموزش داده شده و کوانتیزه شدند، به جز LeNet-5 که توسط Pytorch انجام شد.

۹تحلیل اندازه گیری های RIOT-ML

دراین بخش، نتایج آزمایش هایی را که در بخش [مطالب] توضیح داده ایم،ارائه می دهیم.۸.

۹.۱ارزیابی عملکرد مدل

ارزیابی‌هر مدلمیز۳مصرف منابع مدل LeNet-5 را روی بردهای مختلف‌اینترنت اشیا، که با ارزیابی هر مدل تولید شده اند، نشان می دهد.MCUها بر اساس خانواده گروه بندی شده و به ترتیب صعودی‌فرکانس کلاک در هر گروه مرتب شده اند. MCUهای سری‌ARM Cortex-M هیچ تفاوت قابل توجهی در استفاده از حافظه‌و فضای ذخیره سازی نشان ندادند و با افزایش فرکانس هسته،تأخیر محاسباتی کاهش یافت. یک**پرت**برد rpi-pico مبتنی‌بر RP2040 است، اما 16 کیلوبایت رم اضافی در واقع برای اشکال زدارزرو شده است. با پشتیبانی کامل از پردازش سیگنال دیجیتال(DSP) و مجموعه دستورالعمل های Thumb-2، میکروکنترلرهای‌Cortex-M3 و -M4 با فرکانس کلاک هسته یکسان،عملکرد بهتری نسبت به Cortex-M0+ دارند. یکی دیگر **پرت**روی SiFive RISC-V MCU کشف شد. با بالاترین فرکانس کلاک‌هسته، این MCU از نظر تأخیر محاسباتی و استفاده از حافظه،نامطلوب ترین رتبه را کسب کرد. این MCU از یک فلش خارجی‌رابط جانبی سریال (SPI) NOR برای ذخیره سازی داده ها و برنامه هااستفاده می کند که باعث افت شدید عملکرد در هنگام غیرفعال‌کردن حافظه پنهان می شود.

میز۴نتایج مدل های مختلف ML را روی وظایف نماینده TinyML روی‌بردهای IoT منفرد ارائه می دهد و پشتیبانی جهانی از چارچوب های‌مختلف ML و ساختارهای مدل را به نمایش می گذارد. به‌جز LeNet-5 که روی یک دستگاه میزبان محلی با Pytorch آموزش دیده‌است، بقیه از باغ وحش های مدل متن باز آمده اند. ستون های حافظه‌و ذخیره سازی به میزان مصرف منابع آنها اشاره دارند.

همانطورکه انتظار می رود، مشاهده می کنیم که چگونه مصرف فضای‌ذخیره سازی متناسب با کاهش تعداد پارامترهای مدل کاهش می یابد.یک مورد پرت آشکار این است که‌کوچک *DS-CNN* که فضای ذخیره سازی‌بیشتری نسبت به ... مصرف می کند.*لی نت-۵*اگرچه تقریباً دو برابر پارامترهای مدل دارد. با این حال، تجزیه و تحلیل بیشتر نشان‌داد که‌کوچک *DS-CNN* تقریباً سه برابر فاکتورهای مقیاس بندی بیشتری(در نقطه شناور) در مقایسه با*لی نت-۵*این عوامل مقیاس بندی‌به عنوان پارامترهای مدل محسوب نمی شوند، اما حجم زیادی‌از فضای ذخیره سازی را اشغال می کنند.
همانطورکه انتظار می رفت، پیچیده ترین مدل،*موبایل نت وی۱*، بیشترین حافظه (RAM) و منابع محاسباتی را مصرف کرد.

بااین حال، همانطور که در جدول نشان داده شده است۴سربار اجرا،مصرف حافظه و تأخیر محاسباتی را نمی توان صرفاً از روی مقدارپارامترهای مدل به طور کلی به طور قابل اعتمادی پیش بینی‌کرد. ساختار مدل و الگوی محاسباتی مرتبط با آن، و همچنین‌اثرات فشرده سازی، نیز به شدت بر سربار اجرا تأثیر می گذارند(از این رو، RIOT-ML به عنوان یک ابزار آزمایشی برای محک زنی‌مفید است!).

ارزیابی‌به ازای هر اپراتورما در اینجا از یک مدل کوچک با تنها سه‌لایه از TFlite به عنوان نمونه استفاده کردیم تا از پیچیدگی غیرضروری‌در نمایش جلوگیری کنیم و نتایج خروجی در جدول ارائه‌شده است.۵گلوگاه های محاسباتی در عملگر قرار دارند *add_nn_relu*و*add_nn_relu_1* و با بالاترین میزان مصرف حافظه‌و فضای ذخیره سازی نیز همراه است. می توانیم لایه های متناظرمدل اصلی را با نشانه هایی از پارامترهای مرتبط، که وزن ها،بایاس یا سایر پارامترهای قابل آموزش مدل هستند، ردیابی‌کنیم و اعمال استراتژی های بهینه سازی را روی لایه های هدفمندامکان پذیر سازیم.

۹.۲به روزرسانی های OTA مدل امن

درجدول نشان می دهیم۶تفکیک مصرف منابع. از آنجایی که به روزرسانی های‌کامل و جزئی از یک پشته نرم افزاری مشترک استفاده‌می کنند، مصرف آنها یکسان باقی می ماند. مشاهده می کنیم‌که به دلیل تعداد زیاد پارامترهای مدل ثابت (وزن ها و بایاس)، مدل ML مؤلفه ای است که بیشترین فضای ذخیره سازی(حافظه فلش) را استفاده می کند. در مورد استفاده از حافظه(RAM)، پشته شبکه و ماژول SUIT به دلیل نیاز به بافر زیاد برای‌مدیریت بسته های شبکه و بارهای SUIT، از حافظه نسبتاً بیشتری‌استفاده می کنند. همچنین مشاهده می کنیم که پیاده سازی های‌رمزنگاری به کار رفته توسط SUIT (یعنی libcose و c25519) حداقل مصرف حافظه و فضای ذخیره سازی را دارند.

۹.۳ارزیابی اولیه هزینه های انتقال شبکه

ماعلاوه بر این در جدول اندازه گیری کردیم۶اندازه مانیفست SUIT وبارهای باینری به روزرسانی که باید برای انجام فرآیند به روزرسانی مدل‌از طریق شبکه منتقل شوند. اینها

جدول۵خروجی ارزیابی هر اپراتور از مدل‌سینوس TFlite روی برد stm32f746-disco

اپراتورها	زمان(میکروها)	زمان(%)	پارامز	حافظه	ذخیره سازی
add_nn_relu	۸.۸۵۶	۱۵.۲۲٪	ص۰،ص۱	۰.۱۲۸	۰.۱۲۸
add_nn_relu_1	۴۶.۶۸۲	۸۰.۲۳٪	ص۲،ص۳	۰.۱۲۸	۱.۰۸۸
اضافه‌کردن	۲.۶۴۶	۴.۵۴٪	صفحه۴، صفحه ۵	۰.۰۶۸	۰.۰۶۸

پیشوندتولید شده خودکار uTVM_*tvmgen_default_fused_nn_dense*نام عملگر برای شفاف سازی ارائه نشده است.
میزان‌مصرف حافظه و فضای ذخیره سازی بر حسب کیلوبایت ارائه شده است.

جدول۶مصرف فضای ذخیره سازی و هزینه های‌انتقال به روزرسانی های مدل‌روی برد nrf52840dk با مدل 5-LeNet مستقر شده

نهاد	حافظه	درصد	ذخیره‌سازی (انتقال)	درصد
میکروکنترلر	۲۵۶.۰۰۰	–	۱۰۲۴.۰۰۰	–
میان افزار	۴۳.۸۶۴	۱۰۰.۰۰	۱۲۶.۳۷۴	۱۰۰.۰۰
مدل‌یادگیری ماشینی	۱۲.۴۲۰	۲۸.۳۱	۴۹.۸۵۸	۳۹.۴۵
پشته‌شبکه	۱۳.۷۱۴	۳۱.۲۶	۳۵.۱۲۱	۲۷.۷۹
کت‌و شلوار	۱۳.۹۳۳	۳۱.۷۶	۱۵.۳۵۱	۱۲.۱۴
کریپتو	۱.۳۴۰	۳.۰۵	۵.۸۲۶	۴.۶۱
مانیفستSUIT	–	–	۰.۴۷۱	۰.۳۷
لایه‌نهایی	–	–	۰.۸۴۰	۰.۶۷

دراین اندازه گیری ها، ما رمزگذاری SUIT را لحاظ نکردیم، فقط تأیید امضا و بررسی یکپارچگی را در نظر گرفتیم. مصرف حافظه و فضای‌ذخیره سازی بر حسب کیلوبایت ارائه شده است.

اندازه گیری هامی توانند به سنجش بار وارده بر شبکه کمک کنند. درمورد بارهای داده، اندازه ی میان افزار سریال شده و وزن لایه ی نهایی‌به ترتیب ۱۲۳.۴۴ کیلوبایت و ۰.۸۴ کیلوبایت است. در مقام مقایسه،اندازه ی مانیفست SUIT برابر با ۰.۴۷۱ کیلوبایت است که نسبت‌تقریبی ۲۶۰:۲:۱ را نشان می دهد. این به ویژه بدان معناست که‌سربار بار شبکه ناشی از استفاده از مکانیسم های امنیتی SUIT درRIOT-ML (برای یکپارچگی، احراز هویت و مجوز) هنگام استفاده‌از رویکرد به روزرسانی میان افزار برای به روزرسانی مدل ها،بسیار کمتر از ۱٪ باقی می ماند.

بحث‌در مورد جزئیات به روزرسانی مدلساده ترین رویکرد برای‌به روزرسانی مدل از طریق هوا در دستگاه های اینترنت اشیا، به روزرسانی های‌میان افزار است، زیرا سیستم تعبیه شده از پیچیدگی های‌ناشی از بارگذاری پویا روی سخت افزار ناهمگن رهایی‌می یابد. با این حال، این راحتی با هزینه بالایی از نظر هزینه های‌انتقال شبکه همراه است. اندازه گیری ها در جدول۶ اشاره‌می شود که به روزرسانی فقط مدل (۴۹ کیلوبایت) به جای کل‌میان افزار (۱۲۶ کیلوبایت) بیش از ۶۰٪ در هزینه های انتقال شبکه‌صرفه جویی می کند. فراتر از این، اگر در برخی موارد به روزرسانی های‌مدل بتوانند از یادگیری انتقالی عبور کنند، یعنی به روزرسانی‌فقط بخش هایی از مدل، هزینه های انتقال شبکه نسبتاً ناچیز می شود. به عنوان مثال، در اندازه گیری های ما، ما فقط‌آخرین لایه را به روزرسانی کردیم که بیش از ۹۹٪ از هزینه های انتقال‌شبکه را حذف کرد. با این حال، با این رویکرد، بهبود دقت استنتاج‌با به روزرسانی ممکن است محدودتر باشد. بنابراین، کاربران‌موظفند هنگام طراحی طرح به روزرسانی مدل خود، بین سربارهای‌شبکه، پیچیدگی فنی و دقت مدل تعادل برقرار کنند.

۱۰ RIOT-ML قابل‌تکرار و سفارشی آزمایش ها

ماکد منبع کامل ابزار RIOT-ML را در Github به آدرس زیر منتشرکردیم.https://github.com/TinyPART/RIOT-MLتحت مجوزمتن باز LGPL نسخه ۳. برای جزئیات بیشتر در مورد نحوه شروع‌کار عملی با RIOT-ML، خواننده به راهنمای جامع ارجاع داده می شود.*Readme.md*در مخزن.

ازیک طرف، محققان و متخصصانی که سخت افزار اینترنت اشیادارند که توسط سیستم عامل متن باز RIOT پشتیبانی می شود(در حال حاضر بیش از ۲۵۰ نوع برد، با استفاده از بیش از ۶۰نوع CPU)[۹]می توانند از RIOT-ML به صورت آماده و مستقیماً روی‌بردهای خود استفاده کنند.

ازسوی دیگر، همراه با استفاده از بستر آزمایشی رایگان و آزاد IoT-Lab،[۱۰]حتی محققان و متخصصانی که چنین سخت افزاری را درمحل خود ندارند، می توانند با استفاده از RIOT-ML، کمپین های‌ارزیابی تجربی در مقیاس بزرگ انجام دهند.

دیدگاه هاهمچنان که پشتیبانی از بردها و پردازنده های RIOT به مرور زمان‌گسترش و بهبود می یابد، و همچنان که uTVM نیز به طور موازی پشتیبانی‌از سایر معماری ها را گسترش می دهد (هر دو جامعه متن باز بسیارفعال هستند)، RIOT-ML می تواند در مدت زمان کوتاهی پشتیبانی خودرا برای موارد استفاده جدید گسترش دهد و به طور خودکار پشتیبانی uTVM را برای بردهای جدید و پشتیبانی RIOT را برای مدل های جدید اضافه‌کند. به این ترتیب، RIOT-ML ممکن است به صورت ارگانیک رشد کندو به یک پیوند مفید بین این دو جامعه تبدیل شود.

علاوه‌بر این، اگرچه کار روی RIOT-ML در این مقاله فقط بر استنتاج‌روی میکروکنترلرهای تک هسته ای متمرکز بوده است، اما پتانسیل‌بالایی برای گسترش جعبه ابزار ارائه شده توسط RIOT-ML برای‌پشتیبانی از سناریوهای یادگیری روی دستگاه و بهینه سازی بهره برداری‌از میکروکنترلرهای چند هسته ای وجود دارد.

۱۱نتیجه گیری

این‌مقاله RIOT-ML را معرفی می کند، یک جعبه ابزار جدید که ما برای‌ساده سازی تلاش های متخصصان AIoT طراحی کرده ایم. RIOT-ML ادغام مداوم مدل TinyML، استقرار مداوم ایمن و ارزیابی‌عملکرد را از راه دور بر روی شبکه، روی مجموعه ای از دستگاه های‌IoT ناهمگن تحت آزمایش، تسهیل می کند. به طور دقیق تر،با RIOT-ML، کاربران می توانند خروجی مدل Zoo را توسط‌چارچوب های مختلف یادگیری ماشین سنتی (مانند PyTorch، TensorFlow) دریافت کرده و خودکارسازی کنند.

[۹] دیدنhttps://github.com/RIOT-OS/RIOT/tree/master/boards

[۱۰] ببینیدhttps://www.iot-lab.info/

تطبیق پذیری،استقرار و ارزیابی عملکرد آنها هنگام اجرا بر روی طیف‌گسترده ای از بردهای اینترنت اشیا و کیت های توسعه مبتنی‌بر انواع مختلف میکروکنترلرهای کم مصرف (RISC-V ، ARM Cortex-M، ESP32). فعال سازی و تسهیل چنین ماتریس های‌آزمایشی بزرگی در واقع برای پیشرفت حوزه AIoT بسیارمهم است. برای این منظور، ما همچنین یک پیاده سازی متن بازبسیار قابل استفاده مجدد، مستندسازی شده و قابل تنظیم‌از RIOT-ML منتشر کردیم که از جوامع متن باز پویا مرتبط باRIOT و uTVM بهره می برد. در آخر، ما با ارائه نتایج ارزیابی تجربی‌اولیه در یک بستر آزمایشی با دسترسی آزاد، کاربرد عملی RIOT-ML را به نمایش می گذاریم.

تقدیرنامه هانویسندگان مایلند از سدریک آجی، نجیب آچیر و فلیکس بیسمن برای‌بحث ها و پیشنهادهای مفیدشان تشکر کنند.

سهم‌نویسنده.نرم افزار را توسعه داده و آزمایش کردند. همه نویسندگان مقاله‌را بررسی کردند KS و ZH، KZ متن اصلی مقاله را مفهوم سازی و نوشتند؛EB و ZH

بودجهبودجه دسترسی آزاد توسط Projekt DEAL فراهم و سازماندهی شده است. تحقیقاتی‌که منجر به این نتایج شده است، تا حدودی از برنامه امنیت سایبری آلمانی /فرانسوی MESRI-BMBF تحت توافقنامه اعطای کمک هزینه شماره CYAL-0005 -ANR-20 و 16KIS1395K بودجه دریافت کرده است.

دردسترس بودن داده هادر طول مطالعه فعلی هیچ مجموعه داده ای تولید یا تجزیه و تحلیل‌نشد.

اعلامیه ها

تضادمنافعنویسندگان هیچ گونه تضاد منافعی را اعلام نمی کنند.

سلب‌مسئولیتاین مقاله صرفاً منعکس کننده دیدگاه های نویسندگان است. MESRI و BMBF مسئولیتی در قبال هرگونه استفاده ای که ممکن است از اطلاعات موجود در آن صورت گیرد،ندارند.

منابع

1.سزه وی، چن وای اچ، یانگ تی جی، امر جی اس (2017) پردازش کارآمد شبکه های‌عصبی عمیق: یک آموزش و بررسی. 2329–2295:)12(105 Proc IEEE

2.شارما اچ، حق ای، بلابجرگ اف (2021) یادگیری ماشین در شبکه های حسگربی سیم برای شهرهای هوشمند: یک بررسی. الکترونیک 10(9) 1012:

3.سانچز-ایبورا آر، اسکارمتا ای اف (2020) اشیاء هوشمند مقرون به صرفه با قابلیتTinyML: چالش ها و فرصت ها. مجله مدارهای 18–4:)3(20 IEEE Syst

4.ری پی پی (2022) مروری بر TinyML: جدیدترین ها و چشم اندازها. مجله علوم‌کامپیوتر دانشگاه ملک سعود 34(4):1595–1623

5.هوانگ زد، زندبرگ کی، شلایزر کی، باچلی ای (2023) U-TOE: جعبه ابزار ارزیابی‌جهانی TinyML برای اینترنت اشیا کم مصرف. درسال ۲۰۲۳، دوازدهمین کنفرانس بین المللی ifip/ieee در مورد ارزیابی و مدل سازی‌عملکرد در شبکه های سیمی و بی سیم (pemwn)، IEEE، صفحات۱ تا ۶

6.ساها اس اس، ساندها اس اس، سریواستاوا ام (2022) یادگیری ماشین برای‌سخت افزار کلاس میکروکنترلر - بررسی. 21390–21362:)22(22 J IEEE Sensors

7.هام او، باچلی ای، پیترسن اچ، تیسفتس ان (2015) سیستم های عامل برای‌دستگاه های ارزان قیمت در اینترنت اشیا: یک بررسی. 734–720:)5(3 IEEE Internet Things J

8.باچلی ای، گوندوغان سی، هام او، کیتزمن پی، لندرز ام اس، پترسن اچ، والیش‌ام (2018) RIOT: یک سیستم عامل متن باز برای دستگاه های تعبیه‌شده سطح پایین در اینترنت اشیا. 4440–4428:)6(5 Things J IEEE Internet

9.لیما ال ای، کیمورا بی ال، راستِ وی (2019) محیط های آزمایشی برای اینترنت‌اشیا: یک بررسی. 3211–3203:)9(19 IEEE Sens J

11(:58-67 49)نظرسنجی در مورد امکانات برای تحقیقات .IEEE Commun Mag آزمایشی‌اینترنت اشیا)2011(T D, Mitton N, Razafindralambo

10. Gluhak A, Krco S, Nati M, Pfisterer .صفحات459-464، IEEE، در سال 2015)wfiot(در مورد اینترنت اشیا IEEE در مقیاس‌بزرگ. در: دومین انجمن جهانی IoT یک بستر آزمایشی آزمایشی باز :LAB- FIT IoT و همکاران (2015) Harter G، MittonN، Noel T، Vandaele J ،

11. AdjihC، Baccelli E، FleuryE

۱۲.بنبری سی آر، ردی وی جی، لام ام، فو دبلیو، فاضل ای، هولمن جی، لوخموتوف‌ای و همکاران (۲۰۲۰) ارزیابی سیستم های TinyML: چالش ها ومسیرها. پیش از چاپ درarXiv:2003.04821

13. Banbury C، Reddi VJ، Torelli P، Holleman J، Jeffries N، Kiraly سی،پائو دی و همکاران (2021) معیار کوچک MLPerf. پیش چاپ در arXiv:2106.07597

14.عثمان آ، عبید یو، جما ال، پروتو ام، برونلی دی (2022) بنچمارک پلتفرم هایTinyML. در: کاربردهای الکترونیک در صنعت، محیط زیست وجامعه: Applepies 2021، Springer صفحات 139-148

15. Sudharsan B، Salerno S، Nguyen DD، Yahya M، وحید A، Yadav پی،علی ام آی (۲۰۲۱) بنچمارک TinyML: اجرای شبکه های عصبی کاملاً متصل‌روی میکروکنترلرهای معمولی. در: هفتمین مجمع جهانی IEEE در مورداینترنت اشیا (wf-iot) در سال ۲۰۲۱، IEEE، صفحات ۸۸۳–۸۸۴

۱۶.چن تی، مورو تی، جیانگ زد، ژنگ ال، یان ای، شن اچ، چز ال و همکاران (۲۰۱۸) TVM: یک کامپایلر بهینه سازی سرتاسری خودکار برای یادگیری عمیق،۵۷۸–۵۹۴

17.کیو اچ، واولیدو آی، لی جی، پرگامنت ای، واردن پی، چینچالی اس، کاتی اس(2022) ML-EXray: قابلیت مشاهده در استقرار ml در لبه. 351– Proc Mach Learn Syst 4:337

18.یوسف زاده اصل-میاندوآب ای، روبروک تی، توزون پی (2023) پروفایل بندی و نظارت‌بر وظایف آموزشی یادگیری عمیق. در: مجموعه مقالات سومین کارگاه آموزشی‌یادگیری ماشین و سیستم ها، صفحات 18-25. https://doi.org/10.1145/3578356.3592589

19. Kreuzberger D, Kühl N, Hirschl S)2023(عملیات‌یادگیری ماشین (MLops:) مرور کلی، تعریف و معماری. IEEE Access 11:31866–31879. https://doi.org/10.1109/ACCESS.2023. 3262138

20.لی ام تی، آربل جی (2023) TinyMLops برای MCUهای فوق العاده کم مصرف بلادرنگ‌که برای طبقه بندی رویدادهای مبتنی بر فریم اعمال می شوند. در: مجموعه مقالات‌سومین کارگاه آموزشی یادگیری ماشین و سیستم ها، نیویورک، نیویورک، ایالات‌متحده آمریکا: انجمن ماشین آلات محاسباتی، صفحات 148-153

21.دیاز-دآرکایا جی، توره-باستیدا ای آی، زاراتی جی، میون آر، آلمیدا ای (2023) مطالعه مشترک چالش ها، فرصت ها و نقشه راه MLOps و AIOps: یک بررسی سیستماتیک. 30–1:)4(56 .ACM Comput Surv https://doi.org/10.1145/3625289
22.زندبرگ ک، شلایزر ک، آکوستا ف، شوفنیگ ه، باچلی ای (2019) به روزرسانی های امن‌میان افزار برای دستگاه های اینترنت اشیا محدود با استفاده از استانداردهای باز:بررسی واقعیت. IEEE Access 7:71907–71920
23.متتوا ان اس، تارویری پی، ابو-محفوظ ای ام، آدیگان ام او (2019) به روزرسانی های‌امن میان افزار در اینترنت اشیا: یک بررسی. در کنفرانس‌بین المللی فناوری اطلاعات و مهندسی چندرشته ای 2019 (imitec)، IEEE، صفحات 1-7
24.بانگاس جی، زندبرگ کی، باچلی ای، هرمان ای، اسمیت بی (2022) امنیت‌به روزرسانی نرم افزار مقاوم در برابر کوانتوم در دستگاه های تعبیه شده‌شبکه ای کم مصرف. در کنفرانس بین المللی رمزنگاری کاربردی‌و امنیت شبکه، اشپرینگر، صفحات 872-891
41-35:)2(58 .IEEE Commun Mag به روزرسانی های نرم افزار هوایی در اینترنت‌اشیا: مروری بر اصول کلیدی)2020(I, De Poorter E
25. Bauwens J, Ruckebusch P, Giannoulis S, Moerman
26.بورمن سی، کاستلانی ای پی، شلبی زد (2012) CoAP: یک پروتکل کاربردی‌برای میلیاردها گره کوچک اینترنتی. 67–62:)2(16 Comput IEEE Internet
برای‌نظرات IETF درخواست .)CoAP(پروتکل برنامه محدود
27. ShelbyZ، HartkeK، BormannC)2014(RFC7252:
28.موران بی، شوفنیگ اچ، براون دی، مریاک ام (2021) معماری به روزرسانی میان افزاربرای اینترنت اشیا. RFC 9019.https://doi.org/10.17487/RFC9019
۲۹.هانسن تی سوم، دی ای ای (۲۰۰۶) الگوریتم های هش امن ایالات متحده (SHA و HMACSHA). ۴۶۳۴ RFC.https://doi.org/10.17487/RFC4634
I)2017(الگوریتم امضای دیجیتال منحنی ادواردز)EdDSA(.RFC8032. Josefsson S, Liusvaara 30. RFC8032 /10.17487/ https://doi.org
۳۱.تسوفنیگ اچ، هاسلی آر، موران بی، براون دی، تاکایاما کی (۲۰۲۳) بارهای رمزگذاری‌شده در مانیفست های SUIT (شماره پیش نویس اینترنت، 18-draft-ietf-suit-firmwareencryption). کار در حال انجام. گروه ویژه مهندسی‌اینترنت. برگرفته از https://datatracker.ietf.org/doc/draft-ietfsuit-firmware-encryption/18/
۳۲.سلندر جی، ماتسون جی پی، پالومبینی اف (۲۰۲۴) دیفی-هلمن زودگذر بر فراز۹۵۲۸ RFC .)EDHOC(COSE.https://doi.org/10. 17487/RFC9528/
33.کاهان دبلیو (1996) استاندارد IEEE 754 برای حساب ممیز شناور دودویی. یادداشت های‌سخنرانی وضعیت 11:)1776–94720(754 IEEE
34. Schaad J)2017(CBOR امضا و رمزگذاری شی)COSE(.RFC8152. https://doi.org/10.17487/RFC8152

یادداشت‌ناشراشپرینگر نیچر در مورد ادعاهای مربوط به صلاحیت قضایی در نقشه های‌منتشر شده و وابستگی های سازمانی بی طرف باقی می ماند.