Journal of Applied And Basic Machine Intelligence Research (ABMIR)

پژوهش‌های نظری و کاربردی هوش ماشینی

سال دوم، شماره اول، بهار و تابستان ۱۴۰۳، صفحات ٦٦-٥١

محاسبه مقدار آسیب‌پذیری مدارهای مجتمع دیجیتال در برابر تروجان سخت‌افزاری با استفاده از شبکه‌های عصبی پیچشی

هادی جهانی راد [*1]

گروه مهندسی برق، الکترونیک و مخابرات، دانشکده مهندسی، دانشگاه کردستان، سنندج، ایران

مقاله پژوهشی

تاریخ دریافت:
۱۴۰۲/۱۲/۱۷

تاریخ پذیرش:
۱۴۰۳/٤/١٦

کلیدواژه‌ها:
مدارهای مجتمع دیجیتال، تروجان های سخت افزاری، شبکه‌های عصبی پیچشی، کلاس بندی، یادگیری ماشین

نویسنده‌مسئول:
h.jahanirad@uok.ac.ir

چکیده

با پیشرفت تراشه‌های مجتمع دیجیتال و پیاده‌سازی سیستم‌های پیچیده برروی آن‌ها، مخاطراتی در رابطه با عملکرد آن‌ها ایجاد شده است. تروجان‌های سخت‌افزاری (HT) از مهمترین نوع مخاطرات هستند که سبب ایجاد خطا در عملکرد تراشه، افزایش توان مصرفی و نشت‌کردن اطلاعات ذخیره‌شده برروی تراشه‌ها می‌شوند. درنتیجه، ارزیابی میزان آسیب‌پذیری تراشه‌ها در برابر انواع مختلف تروجان‌های سخت‌افزاری دارای اهمیت بسیار زیادی است. در این مقاله روشی دقیق در سطح چیدمان (layout)، برمبنای استفاده از شبکه‌های عصبی پیچشی (CNN)، برای محاسبه میزان آسیب پذیری تراشه‌های دیجیتال در برابر تروجان‌های سخت‌افزاری ارائه شده است. عوامل اصلی مؤثر بر میزان خطرپذیری تراشه‌های دیجیتال شامل میزان فضاهای خالی در چیدمان، منابع مسیردهی استفاده نشده، فعالیت سیگنال‌های داخلی، وقابلیت آزمون-پذیری گیت‌های مدار می‌باشند. برای تولید مجموعه داده مناسب، چیدمانِ فیزیکیِ هر پیاده‌سازی از یک مدار دیجیتال با استخراج این عوامل، به یک تصویر دیجیتال تبدیل شده است. پس از تولید مجموعه داده مناسب که شامل ۱۰۰۰۰ تصویر است، فرآیند یادگیری شبکه عصبی پیچشی تکمیل می-شود و شبکه آموزش یافته برای تعیین میزان آسیب‌پذیری در برابر تروجان-های سخت‌افزاری مورد استفاده قرار می‌گیرد. با مشخص‌شدن میزان آسیب-پذیری مدار پیاده‌سازی شده، طراح می‌تواند تغییرات لازم را برای مقاوم کردن تراشه در برابر تروجان‌های سخت‌افزاری اعمال نماید. نتایج شبیه‌سازی برروی چیدمان مدارهای معیار (ISCAS 85, 89) نشان می‌دهد، میزان دقت رهیافت پیشنهادی ۹۲٪ می‌باشد. همچنین روش پیشنهادی، مشکل ناشی از عدم مدلسازی دقیق عوامل مؤثر در تعیین خطرپذیری در روش‌های پیشین را مرتفع کرده و دقت محاسبه آسیب‌پذیری را نسبت به بهترین روش موجود در مطالعات پیشین، ۱۷٪ افزایش می‌دهد.

: 10.22034/ABMIR.2024.21351.1049

E-ISSN: 2821-2037

۱- مقدمه

جایگذاری تروجان‌های سخت‌افزاری[1] در مدارهای مجتمع مدرن، برای ایجاد اختلال در عملکرد سیستم و یا مصرف توان بیشتر، به یکی از مخاطرات جدی تبدیل شده است. به‌طور کلی تروجان سخت‌افزاری، یک هسته یا ماجول سخت‌افزاری کوچک است که در مکان‌های خاصی از چیدمان IC[2] قرارداده می‌شود. فعالسازی تروجان‌های سخت‌افزاری با یک تریگر که متصل به گره‌های با میزان فعالیت کم هستند، انجام می‌شود. آشکارسازی وجود تروجان‌های سخت‌افزاری در تراشه‌های مورد استفاده در سیستم‌های حساس (نظامی، مخابراتی و مانند آن) بسیار با اهمیت است. روش‌های آشکارسازی، به سه دسته اصلی تقسیم می‌شوند. دسته اول از روش‌های آشکارسازی، بر مبنای آزمون‌های منطقی عمل می‌کنند و برای آشکارسازی تروجان‌های همیشه-فعال، کارایی بیشتری دارند. دسته دوم از روش‌های آشکارسازی، به تحلیل اثرات جانبی (side channel) تروجان سخت‌افزاری، مانند بوجود آمدن نقاط داغ[3] در اثر فعالیت تروجان، افزایش توان مصرفی تراشه و یا افزایش تأخیر مسیر بحرانی مدار، می‌پردازند. راهکارهایی که در این دسته قرار می‌گیرند، برمبنای تحلیل الگوی حرارتی تراشه، الگوی توان مصرفی تراشه، و ابزارهای یادگیری ماشین استوارند. دسته سوم براساس مهندسی معکوس تراشه دیجیتال به آشکارسازی تروجان‌های سخت‌افزاری می‌پردازند. این دسته از روش‌ها، از جمله روش‌های مخرب بوده و تنها در موارد خاصی برای تولید تراشه‌های طلایی مورد استفاده قرار می‌گیرند.

تعدادی از پژوهش‌های مهم در زمینه آشکارسازی تروجان‌های سخت‌افزاری به‌شرح زیر است. در مرجع [۱] از روش مهندسی معکوس در سطح چیدمان و گیت، برای آشکارسازی استفاده‌شده است که بسیار زمان‌بر و پرهزینه است. روشی دیگردر مرجع [۲] بر مبنای قرار دادن حسگرهای حرارتی در بخش‌های مختلف IC ارائه شده است که در زمان اجرا (runtime) با استفاده از اطلاعات حسگرها وجود تروجان‌ها را آشکار می‌کند. این روش پرهزینه است و دقت آن به‌شدت وابسته به دقیق بودن اطلاعات حسگرها می‌باشد. He و همکارانش در مرجع [۳] از تحلیل اثرات جانبی الکترومغناطیسی ناشی از تروجان‌های سخت‌افزاری برای آشکارسازی استفاده کرده‌اند.

در مرجع [٤] از اطلاعات فاز در اسکن حرارتی فروسرخ برای ایجاد الگوی حرارتی تراشه استفاده‌شده است. با بهره‌گیری از این روش، نقاط داغ با ابعاد کوچک (که می‌توانند ناشی از فعالیت تروجان‌ها باشند) قابل تشخیص هستند. Koushanfar و Mirhosseini در مرجع [٥] یک روش چند حالته با ترکیب الگوهای حرارتی و توانی برای آشکارسازی توسعه داده‌اند. در مرجع [٦] از روش تحلیل الگوی دمایی (که با تبدیلات لازم از الگوی توانی استخراج می‌شوند) برای آشکارسازی استفاده‌شده است.

از پردازش تصویر برای آشکارسازی تروجان‌های سخت‌افزاری در پژوهش‌های مختلف استفاده‌شده است. تصویربرداری SEM از سطح تراشه و تحلیل آن برای آشکارسازی تروجان سخت‌افزاری براساس روش‌های پردازش تصویر، در مرجع [۷] ارائه شده است. Rahimifar و Jahanirad در مرجع [۸] از روش‌های پردازش تصویر برای تحلیل الگوی حرارتی مدار مجتمع در راستای آشکارسازی تروجان‌های سخت‌افزاری استفاده کرده‌اند.

با توجه به تنوع بسیار زیاد تروجان‌های سخت‌افزاری و عدم توانایی کامل روش‌های آشکارسازی برای تشخیص وجود آن‌ها در تراشه نهایی، ضروری است که طراحی مدار مجتمع به‌گونه‌ای انجام شود که امکان قراردادن تروجان‌های سخت‌افزاری در آن تراشه به حداقل ممکن برسد. برای رسیدن به این هدف، باید ارزیابی میزان آسیب‌پذیری چیدمان طراحی شده IC در مقابل تروجان سخت‌افزاری انجام گیرد. ویژگی‌های متعددی در این ارزیابی موثرند که مهمترین آن‌ها به شرح زیر است: معمولاً طراحان تروجان‌های سخت‌افزاری، هسته سخت‌افزاری خود را در فضاهای خالی از چیدمان تراشه، قرار می‌دهند. از طرف دیگر اتصالات لازم برای تریگر کردن و همچنین اعمال خروجی هسته‌های

[1] Hardware Trojans
[2] Integrated Circuits
[3] Hotspots

تروجان‌های سخت‌افزاری به بخش‌های دیگر مدار مجتمع، از طریق منابع استفاده نشده در شبکه مسیردهی تراشه انجام می‌گیرد. تریگر کردن هسته‌های سخت‌افزاری تروجان‌ها از طریق اتصال به گره‌هایی از مدار که فعالیت سیگنالی پایین دارند، انجام می‌شود. این کار به طراحان تروجان‌های سخت‌افزاری کمک می‌کند که از روش‌های آشکارسازی بر مبنای آزمون‌های منطقی، مصون بمانند. همچنین برای عبور از آزمون‌های بر مبنای تأخیر مسیر، تروجان‌ها باید به‌گونه‌ای در تراشه قرار گیرند که تأخیر مسیر بحرانی را افزایش ندهند. بنابراین وجود مسیرهای غیر بحرانی زیاد در ساختار مدار مجتمع می‌تواند آن را در برابر تروجان‌های سخت‌افزاری آسیب‌پذیرتر کند.

با توجه به وجود عوامل متعدد در ارزیابی آسیب‌پذیری مدارهای مجتمع دیجیتال در برابر تروجان‌های سخت‌افزاری، توسعه دادن یک مدل فراگیر و کارا که بتواند انواع مختلف حالت‌های ممکن را پوشش دهد، بسیار پیچیده است. در روش‌های پیشین [۹-۱۱] ابتدا مقادیر فضای خالی، تراکم منابع مسیردهی، فعالیت سیگنال در گره‌های داخلی مدار محاسبه می‌شود. سپس با استفاده از روابط ساده ریاضی (ضرب کردن مقادیر بدست آمده در مرحله اول) میزان آسیب‌پذیری بخش‌های مختلف مدار محاسبه می‌شود. مهم‌ترین اشکال این روش‌ها آن است که ترکیب معیارهای مختلف، به‌صورت ناقص انجام‌شده است و همان‌طور که در بخش‌های بعدی نشان داده می‌شود این امر به نتایج نادرست می‌انجامد.

در این مقاله روشی کارا برای ارزیابی آسیب‌پذیری مدارهای مجتمع دیجیتال در برابر تروجان‌های سخت‌افزاری ارائه می‌شود که در آن ابتدا سطح چیدمان مدار به صورت یک شبکه توری (با سوراخ‌های مربعی شکل) بخش‌بندی می‌شود. سپس برای هرکدام از بخش‌ها معیارهای اصلی محاسبه می‌شود و ترکیب آن‌ها منجر به تولید یک تصویر دیجیتال برای اعمال به شبکه عصبی پیچشی می‌گردد. استفاده از ظرفیت یادگیری بالای شبکه‌های عصبی کانولوشن منجر به یادگیری روابط و قضایای پیچیده بین عوامل مؤثر در میزان خطرپذیری تراشه‌های دیجیتال خواهد شد و درنتیجه نقص‌های مدلسازی در روش‌های پیشین به صورت مناسب برطرف می‌گردد.

رئوس نوآوری‌های این پژوهش را می‌توان به صورت ذیل خلاصه کرد:

- تولید مجموعه داده مناسب و ارائه روش تولید آن برای مطالعات مربوط به بررسی خطرپذیری مدارهای دیجیتال.
- ارائه راهکار مناسب برای استخراج ویژگی‌های اصلی و مؤثر در تعیین میزان خطرپذیری برای مدارهای دیجیتال.
- توسعه یک چارچوب نرم‌افزاری برای تعیین کلاس خطرپذیری در برابر تروجان‌های سخت‌افزاری در مدارهای دیجیتال برمبنای شبکه‌های عصبی پیچشی.

در ادامه مقاله مطالب زیر ارائه می‌شوند. در بخش دوم مفاهیم اولیه برای ورود به بحث توضیح داده می‌شود. در بخش سوم، رهیافت پیشنهادی برای ارزیابی آسیب‌پذیری مدارهای مجتمع دیجیتال در سطح چیدمان با استفاده از شبکه‌های عصبی کانولوشن ارائه می‌شود. نتایج شبیه‌سازی و مقایسه با روش‌های [۹-۱۱] در بخش ٤ ارائه می‌شود و سپس در بخش نتیجه‌گیری جمع بندی مقاله انجام می‌گردد.

۲- محاسبه ویژگی‌های چیدمان تراشه

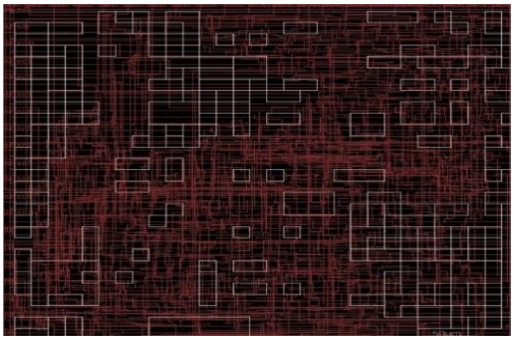
همان‌گونه که در بخش پیش توضیح داده شد، گام اول در ارزیابی آسیب‌پذیری مدار مجتمع در برابر تروجان‌های سخت‌افزاری، محاسبه ویژگی‌های مؤثر در قراردادن هسته‌های تروجان سخت‌افزاری در چیدمان تراشه است. این ویژگی‌ها شامل فضاهای خالی چیدمان، تراکم منابع مسیردهی، کنترل‌پذیری و فعالیت سیگنال مربوط به گره‌های مدار و مسیرهای غیربحرانی (از لحاظ تأخیر) می‌باشد. در زیربخش‌های آینده، نحوه محاسبه هرکدام از این ویژگی‌ها توضیح داده می‌شود.

۲-۱ فضاهای خالی در چیدمان تراشه

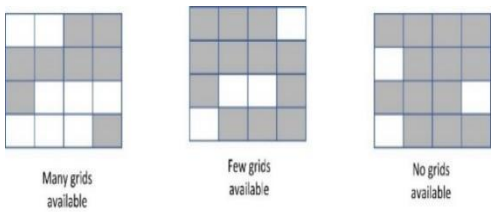
برای تولید چیدمان مدار مجتمع، ابتدا کد HDL[1] مدار در سطح گیت سنتز می‌شود و پس از آن نرم‌افزار Cadence Design System، با استفاده از گیت‌های معیار (کتابخانه‌ای) مدار سنتز شده را به چیدمان بهینه تبدیل می‌کند. به‌صورت کلی، چیدمان نهایی مدارهای مجتمع به یک شبکه منظم M×M از واحدهای مربع شکل

[1] Hardware Description Language

تقسیم‌بندی می‌شود. به‌عنوان مثال چیدمان بهینه مدار معیار b15 یک سطح مربعی 300µm × 300µm است. بنابراین هرکدام از واحدهای شبکه با انتخاب M=10، یک سطح مربعی 30 µm ×30 µm می‌باشد. با فرض اینکه حداقل مساحت خالی که می‌تواند بعنوان فضای خالی درنظر گرفته شود، برابر مساحت چیدمان یک گیت وارونگر (مثلاً INVX0 در کتابخانه SAED-EDK90nm) باشد، می‌توان توزیع فضاهای خالی را در چیدمان نهایی استخراج کرد. در شکل ۱، این توزیع در مورد مدار معیار EthernetMAC10GE نشان داده‌شده است. معمولاً هسته‌های سخت‌افزاری تروجان‌های سخت‌افزاری مساحتی بیشتر از مساحت یک واحد فضای خالی (Unit-Space) نیاز دارد. بعنوان مثال فرض کنید یک هسته سخت‌افزاری تروجان، به مساحتی ۵ برابر Unit-Space نیاز دارد. در شکل ۲، سه واحد مربع شکل از چیدمان تراشه نشان داده‌شده است که در آن‌ها فضاهای خالی به صورت‌های مختلف توزیع شده‌اند. با توجه به تعداد فضاهای خالی مورد نیاز برای پیاده‌سازی تروجان‌های سخت‌افزاری، تنها واحد مربع شکل در شکل سمت چپ برای قرار دادن تروجان‌های سخت‌افزاری موردنظر قابل استفاده می‌باشد.



شکل (۱): فضاهای خالی در چیدمان مدار EthernetMAC10GE



شکل (۲): میزان سختی واردکردن تروجان به مدار در شرایط مختلف

۲-۲ تراکم منابع مسیردهی در چیدمان تراشه

همان‌گونه که پیشتر اشاره شد، هسته‌های سخت‌افزاری مربوط به تروجان‌های سخت‌افزاری برای فعال شدن و تأثیر مخرب بر عملکرد مدار نیازمند استفاده از منابع مسیردهی هستند. اطلاعات مربوط به منابع مسیردهی موجود در چیدمان تراشه، از فایل‌های خروجی نرم‌افزار Layout Editor مانند Design Exchange Format و Library Exchange Format استخراج می‌شوند. درنهایت تراکم هر کدام از واحدهای مربع شکل در چیدمان نهایی تراشه، با استفاده از رابطه ۱ محاسبه می‌شود. در این رابطه $U_r(m,n)$ و $A_r(m,n)$ به ترتیب مقدار منابع مسیردهی استفاده‌شده و تعداد منابع مسیردهی موجود در واحد (m,n) هستند. هرچه تعداد سیم‌های بخش مسیردهی در فضای بالایی واحد مربع شکل بیشتر باشد، امکان پیدا کردن منابع استفاده نشده برای تریگر کردن و اتصال خروجی‌ها بیشتر می‌شود.

$$Cong(m,n) = U_r(m,n) \Big/ A_r(m,n) \qquad (۱)$$

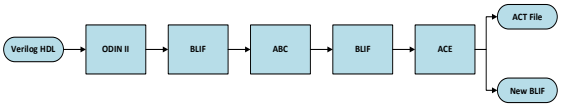
۲-۳ فعالیت سیگنال در گره‌های مدار

تریگر کردن هسته‌های تروجان سخت‌افزاری با استفاده از گره‌هایی از مدار انجام می‌شود که فعالیت پایین‌تری دارند. منظور از فعالیت سیگنال در یک گره، تعداد متوسط 0 و 1 شدن‌های سیگنال در یک بازه زمانی مشخص است. در تحلیل مدارهای مجتمع دیجیتال معمولاً برای بیان فعالیت سیگنال از مفهوم چگالی انتقال سیگنال استفاده می‌شود که طبق رابطه ۲ تعریف می‌شود. در این رابطه $n_x(T)$ تعداد انتقال از 0 به 1 و از 1 به 0 در گره x از مدار در مدت زمان T می‌باشد.

$$D(x) = \lim_{T\to\infty} \frac{n_x(T)}{T} \qquad (۲)$$

در روش پیشنهادی این مقاله، محاسبه چگالی انتقال سیگنال مطابق روند نشان داده در شکل (۳) انجام می‌شود. در مرحله اول، کد Verilog مدار موردنظر وارد سنتزگر ODIN II می‌شود که

خروجی آن یک فایل در قالب BLIF[1] می‌باشد. در مرحله دوم، فایل تولیدشده، توسط نرم‌افزار ABC[2] مورد پردازش قرار می‌گیرد که هدف از آن انجام بهینه‌سازی و نگاشت مدار سنتز شده بر روی تراشه با استفاده از تکنولوژی ساخت موردنظر می‌باشد. خروجی مرحله دوم که در قالب فایل BLIF می‌باشد با استفاده از نرم‌افزار ACE 2 جهت استخراج مقدار چگالی انتقال (فعالیت) گره‌های داخلی مدار، مورد ارزیابی قرار می‌گیرد. نتایج این مرحله در فایل ACT ذخیره می‌گردد.



شکل (۳): مراحل به دست آوردن فایل مربوط به فعالیت سوییچ‌ها

۲-۴ محاسبه تأخیر مسیرها

برای محاسبه تأخیر مسیرهای موجود از ورودی‌ها به خروجی‌های مدار، ابتدا گراف جهت‌دار مدار تولیدشده و با استفاده از ویژگی‌های الکترونیکی گیت‌ها، به هرکدام از یال‌های گراف مقدار عددی تأخیر انتساب داده می‌شود. تأخیر هر مسیر از ورودی به خروجی، با جمع کردن تأخیر یال‌های متعلق به آن مسیر (d_e) محاسبه می‌شود (رابطه ۳). در رابطه (۳)، P نمایانگر مسیر موردنظر و d_e بیانگر تأخیر مربوط به جزء e از مسیر P می‌باشد.

$$delay(P) = \sum_{e \in P} d_e \qquad (۳)$$

وقتی تروجان سخت‌افزاری به یک مسیر از مدار اضافه می‌شود، مقدار تأخیر ورودی به خروجی هسته تروجان سخت‌افزاری به تأخیر مسیر موردنظر اضافه می‌شود. در صورتی که این افزایش تأخیر، مسیر غیر بحرانی را به مسیر بحرانی مدار تبدیل کند، آنگاه وجود تروجان بوسیله روش‌های آزمون برمبنای تأخیر آشکار می‌شود. لذا طراح تروجان‌های سخت‌افزاری سعی در شناسایی مسیرهای غیر بحرانی مداردارد. محاسبه معیار تأخیر مسیر در هرکدام از واحدهای مربع شکل، با شمارش تعداد مسیرهای غیربحرانی در آن واحد انجام می‌شود.

۲-۵ کنترل‌پذیری گره‌های مدار

تریگر کردن تروجان و اتصال آن به سایر بخش‌های مدار، باید در گره‌هایی صورت گیرد که کنترل پذیری و مشاهده‌پذیری کمی داشته باشند. این امر سبب می‌شود که روش‌های آشکارسازی برمبنای آزمون، با احتمال کمتری امکان تشخیص هسته‌های تروجان را داشته باشند. برنامه SCOAP[3] که به‌صورت گسترده در آزمون مدارهای دیجیتال به‌کار برده می‌شود، برای محاسبه میزان کنترل‌پذیری گره‌های داخلی مدار، مورداستفاده قرار می‌گیرد [۱۲]. در این روش، مدار سطح‌بندی می‌شود و با شروع از ورودی‌های اصلی مدار محاسبه کنترل‌پذیری گیت‌های منطقی با توجه به مقدار کنترل‌پذیری ورودی‌های آن گیت انجام می‌شود. منظور از کنترل‌پذیری 0 (CC0[4]) و کنترل‌پذیری 1 (CC1[5]) آن است که ایجاد منطق 0 و 1 در خروجی گیت با استفاده از تنظیم منطق ورودی‌های گیت، به چه اندازه مشکل است. در جدول ۱ نحوه محاسبه CC0 و CC1 برای تعدادی از گیت‌های پایه‌ای ارائه شده است. بعد از اعمال SCOAP برروی مدار دیجیتال و تعیین مقدار کنترل‌پذیری گره‌های مختلف مدار، می‌توان برای واحدهای مربع شکل چیدمان تراشه معیار کنترل‌پذیری را با میانگین گرفتن از مقادیر کنترل‌پذیری گره‌های موجود در آن واحد محاسبه کرد.

جدول (۱): نحوه محاسبه کنترل‌پذیری چهار گیت منطقی

	CC0	CC1
AND	min(CC0(x1), CC0(x2))+1	[CC1(x1)+CC1(x2)]+1
OR	[CC0(x1)+CC0(x2)]+1	min(CC1(x1), CC1(x2))+1
XOR	min[(CC0(x1)+CC0(x2),(CC1(x1)++CC1(x2))]+1	min[(CC0(x1)+CC1(x2),(CC1(x1)++CC0(x2))]+1
BUFFER	CC0(x)+1	CC1(x)+1

[1] Berkeley Logic Interchange Format
[2] A System for Sequential Synthesis and Verification
[3] Sandia Controllability Analysis Program
[4] Combinational Controlability-0
[5] Combinational Controlability-1

۳- روش پیشنهادی برای محاسبه میزان خطرپذیری تراشه‌های دیجیتال دربرابر تروجان‌های سخت‌افزاری

با توجه به وجود عوامل متعدد، تعیین میزان آسیب‌پذیری یک مدار مجتمع در برابر تروجان‌های سخت‌افزاری امری بسیار پیچیده است. شاخص‌ترین روش‌های موجود برای تعیین میزان خطرپذیری، در مراجع [۹-۱۱] ارائه شده‌اند. در مرجع [۱۰]، ابتدا طبق رابطه ٤، مقدار خام خطرپذیری برای هر واحد مربعی شکل از سطح تراشه ($V(r)$) با ضرب کردن مقدار نرمالیزه‌شده فضای خالی ($WS(r)$) و تعداد منابع مسیردهی بلااستفاده ($UR(r)$)، محاسبه می‌شود. سپس آسیب‌پذیری واحد مربع شکل r، براساس اینکه تروجان سخت‌افزاری نسبت به تأخیر مقاوم باشد ($V_{Td}(r)$) یا نسبت به توان دینامیک مقاوم باشد ($V_{LP}(r)$) ویا هردو عامل ($V_{Td\&LP}(r)$) با استفاده از روابط ۵-۷ محاسبه می‌شود.

$$V(r) = WS(r) \times UR(r) \quad (٤)$$

$$V_{Td}(r) = V(r) \times N_{NC}(r) \quad (٥)$$

$$V_{LP}(r) = V(r) \times N_{LP}(r) \quad (٦)$$

$$V_{Td\&LP}(r) = V(r) \times N_{NC\&LP}(r) \quad (٧)$$

در روابط (۵) تا (۷)، $N_{NC}(r)$ تعداد مسیرهای غیربحرانی از لحاظ تأخیر در واحد مربعی r، $N_{LP}(r)$ تعداد گره‌هایی است که مقدار نرخ سوئیچینگ آن‌ها از یک مقدار آستانه کمتر است و $N_{NC\&LP}(r)$ تعداد گره‌هایی هستند که مقدار فعالیت سوئیچینگ آن‌ها کمتر از مقدار آستانه می‌باشد و در عین‌حال برروی مسیرهای غیربحرانی قرارگرفته‌اند.

این روش دارای اشکالاتی اساسی در تعیین درست میزان آسیب‌پذیری مدار مجتمع می‌باشد. یکی از مهمترین اشکالات در این روش مربوط به واحدهایی است که یا کاملاً خالی هستند و یا سطح بسیار کمی از آن‌ها اشغال شده است. این واحدها معیار فضای خالی را بخوبی ارضا می‌کنند و مکان مناسبی برای قراردادن هسته‌های سخت‌افزاری تروجان‌های سخت‌افزاری هستند. اما در روش [۱۰]، در صورتی‌که منبع مسیردهی بلااستفاده‌ای در آن واحد قرار نگیرد ($UR(r)=0$) و یا گره‌ای از مدار که داخل واحد است دارای فعالیت سیگنالی بیش از آستانه موردنظر باشد ($N_{LP}(r)=0$)، مقدار آسیب‌پذیری این نوع واحدها 0 تلقی می‌شود. این نتیجه به‌صورت آشکار درست نیست: طراح تروجان سخت‌افزاری می‌تواند هسته تروجان‌های سخت‌افزاری را در فضای خالی موجود درواحد موردنظر قرار دهد و برای تریگر کردن آن از گره‌های مدار در واحدهای کناری که دارای فعالیت سیگنال پایین‌تری هستند، استفاده کند. بنابراین وجود فضای خالی بزرگ در یک واحد می‌تواند معیار خطرپذیری بالایی باشد و صرفاً بدلیل عدم وجود تریگر مناسب نمی‌توان از خطر وجود چنین واحدهایی صرفنظر کرد.

خطای دیگری که در روش بر مبنای معادله اتفاق می‌افتد، مربوط به واحدهای پرتراکم در چیدمان تراشه می‌باشد. این واحدها که معمولاً در مرکز چیدمان قرار می‌گیرند، دارای منابع مسیردهی زیادی هستند و همچنین به‌دلیل قرارگرفتن تعداد بیشتری گره مدار در آن‌ها، احتمال وجود گره‌هایی با فعالیت سیگنال پایین (جهت تریگر کردن تروجان‌های سخت‌افزاری) زیاد است. در این‌صورت، حتی اگر فضای کوچکی از سطح آن‌ها خالی باشد ($WS(r)$ کوچک)، آسیب‌پذیری نهایی این واحدها به‌دلیل بسیار بزرگ بودن سایر معیارها تعداد منابع مسیردهی بلااستفاده ($UR(r)$) و ($N_{LP}(r)$)، مقدار قابل ملاحظه‌ای است. این درحالی است که بدلیل عدم وجود فضای خالی کافی در چنین واحدهایی، اصولاً امکان قراردادن هسته‌های تروجان‌های سخت‌افزاری در آن‌ها وجود ندارد و درنتیجه این واحدها آسیب‌پذیر نیستند.

در رهیافت ارائه شده در [۹] برای محاسبه میزان خطرپذیری تراشه‌های دیجیتال در برابر تروجان‌های سخت‌افزاری، نقشه خطرپذیری تروجان[1] (TVM) برای سطح مشبک چیدمان فیزیکی[2] استخراج‌شده است. TVM در واقع یک تصویر خاکستری با تعداد پیکسل‌های برابر با تعداد مربع‌های واحد موجود در چیدمان فیزیکی مشبک است. میزان خاکستری بودن هر پیکسل نمادی از

[1] Trojan Vulnerability Map

[2] Layout

میزان خطرپذیری آن است. مقدار سطح خاکستری هر پیکسل بر اساس معیار TISF[1] و مطابق رابطه (۸) محاسبه می‌شود.

در این رابطه BWS نمایانگر نسبت فضای خالی در واحد مورد نظر، TWS نسبت فضای خالی کل چیدمان فیزیکی، BFF و TFF به ترتیب بیانگر تعداد DFF در واحد مورد نظر و در کل چیدمان فیزیکی می‌باشند. WSD بیانگر توزیع فضاهای خالی است که با استفاده از رابطه (۹) محاسبه می‌شود. در این رابطه n، (x_i,y_i) و (x_m,y_m) به ترتیب بیانگر تعداد واحدهای فضای خالی موجود در واحد مربعی مورد بررسی، مختصات مرکز واحد مربعی و مختصات مرکز واحد فضای خالی می‌باشند.

$$TISF = \frac{\frac{BWS}{TWS} \times \frac{BFF}{TFF}}{WSD} \quad (۸)$$

$$WSD = \sum_{i=1}^{n} \frac{1}{n}[(x_i - x_m)^2 + (y_i - y_m)^2] \quad (۹)$$

اشکالات مربوط به مدلسازی فضاهای خالی روش ارائه شده در مرجع [۱۰]، در این رهیافت نیز وجود دارد. همچنین تمرکز این روش برروی معیارهای فضای خالی و تعداد فلیپ‌فلاپ‌های داده (DFF)، منجر به عدم بررسی سایر معیارهای مهم از جمله منابع مسیردهی، میزان فعالیت سیگنال و غیره شده‌است و بنابراین این رهیافت دقت متوسطی در تعیین میزان خطرپذیری چیدمان فیزیکی دارد.

در مرجع [۱۱] رهیافت دیگری برای تعیین میزان خطرپذیری در سطح چیدمان فیزیکی ارائه شده است که از سه معیار مختلف استفاده می‌کند. معیار اول بر مبنای استخراج هیستوگرام فضاهای خالی چیدمان فیزیکی می‌باشد که در آن ارتباط بین واحدهای فضای خالی با استفاده از روش ٤ همسایگی شناسایی می‌شود. معیار دوم مربوط به توانایی بلوکه کردن دسترسی به گره‌های با مقدار فعالیت پایین است. همان‌طور که پیشتر اشاره شد، این گره‌ها می‌توانند به‌عنوان تریگر برای هسته تروجان مورد استفاده قرارگیرند. نویسندگان معتقدند که با افزایش تراکم مسیرها در اطراف این گره‌ها، منابع مسیردهی بلااستفاده که امکان دسترسی طراح تروجان به این گره‌ها را ایجاد می‌کنند، به شدت کاهش می‌یابند. معیار سوم به فاصله بین گره‌های دارای فعالیت سیگنال پایین با فضاهای خالی موجود در سطح چیدمان فیزیکی مربوط می‌شود. بدیهی است که هرچه این فاصله کمتر باشد، امکان ساختن هسته تروجان و اتصال آن به تریگر آسان‌تر خواهد بود. از جمله اشکالات این رهیافت عبارت است از عدم مدلسازی دقیق فضاهای خالی، مدلسازی ناقص تریگرها و عدم شناسایی دقیق مسیرهای ممکن برای اتصال تریگر به هسته تروجان.

۳-۱ تولید مجموعه داده مناسب از مدارهای دیجیتال

برای تولید مجموعه داده موردنظر این پژوهش، از مجموعه مدارهای معیار ISCAS 85[2] و ISCAS 89 استفاده‌شده است که شامل ۲۵ مدار دیجیتال با اندازه و ساختارهای مختلف می‌باشند. هرکدام از مدارهای موجود در این مجموعه از تعداد متنوعی گیت منطقی (AND, OR, DFF, ...) تشکیل شده است که در قالب یک گراف چند لایه به هم متصل شده‌اند. همچنین ارتباط اجزاء داخلی مدار از طریق ورودی‌های اصلی (PI) و خروجی‌های اصلی (PO) انجام می‌پذیرد. در هنگام پیاده‌سازی مدارهای دیجیتال بر روی سطح تراشه‌ها، ابتدا الگوی جانشانی مناسبی برای قراردادن گیت‌های منطقی و PI ها و PO ها تولید می‌شود. سپس در مرحله مسیردهی ارتباط بین اجزاء داخلی با استفاده از منابع مسیردهی موجود در تراشه (سیم‌ها، کنتاکت‌ها و Via ها و سوئیچ‌ها) تحقق پیدا می‌کند.

با توجه به تعداد محدود مدارهای دیجیتال موجود در مجموعه‌های ISCAS 85 و ISCAS 89، برای هرکدام از مدارها، چهارصد پیاده‌سازی مختلف تولیدشده است، که درنهایت منجر به تولید ۱۰۰۰۰ پیاده‌سازی با ویژگی‌های متفاوت شده است.

در فرآیند ایجاد مجموعه‌داده موردنیاز این پژوهش، برای تولید پیاده‌سازی‌های مختلف از یک مدار معیار، مشخصه‌های مختلفی به‌صورت اتفاقی انتخاب‌شده‌اند. ابتدا، مساحت تراشه دیجیتال برای هرپیاده‌سازی از مقدار کمینه مورد نیاز تا ده برابر این مقدار کمینه، تغییر می‌کند. این رویکرد منجر به تولید فضاهای خالی با اندازه‌ها و الگوهای متفاوت در پیاده‌سازی‌های یک مدار می‌گردد. سپس،

[1] Trojan Insertion Simplicity Factor

2 International Symposium of Circuits and Systems

برای هر پیاده‌سازی، جانشانی منحصر به فردی درنظر گرفته می‌شود. به عبارت دیگر در ۴۰۰ پیاده‌سازی مختلف یک مدار معیار، ۴۰۰ جانشانی متفاوت تولید می‌شود. به دلیل متفاوت بودن مقدار فعالیت سیگنال، کنترل‌پذیری و مشاهده‌پذیری گیت‌های مختلف، این رویکرد موجب تولید ویژگی‌های مختلف برای یک موقعیت هندسی خاص از سطح تراشه خواهد شد. درنهایت، در هر پیاده‌سازی از یک مدار معیار، انتخاب نوع خاصی از طراحی برای پیاده‌سازی گیت‌های منطقی از میان طرح‌های مختلف (طراحی مکملی، نسبی، ترانزیستور گذر و دروازه انتقالی) صورت می‌گیرد. با توجه به متفاوت بودن اندازه و چیدمان فیزیکی گیت‌های منطقی در این روش‌های طراحی، توزیع و الگوی فضاهای خالی و میزان تراکم در فرایند مسیردهی در مکان‌های مختلف در سطح تراشه، از تنوع زیادی برخوردار خواهد شد.

مرحله دوم از فرآیند تولید مجموعه‌داده، شامل تبدیل هرکدام از پیاده‌سازی‌های تولیدشده برای مدارهای معیار، به یک تصویر رنگی (RGB) جهت استفاده در مراحل یادگیری و آزمون شبکه‌های عصبی عمیق می‌باشد. این تصاویر دارای ابعاد ۱۲۸×۱۲۸ پیکسل هستند. مرحله اول در این تبدیل عبارت است از مشبک کردن سطح تراشه به واحدهای مربع شکل به‌گونه‌ای که هرکدام از این واحدها به یک پیکسل از تصویر نگاشته می‌شود. در تصویر رنگی، هرکدام از پیکسل‌ها با سه مؤلفه قرمز، سبز و آبی مشخص می‌شود و با یک عدد N بیتی نمایش داده می‌شود. به عنوان مثال اگر N=8 باشد، آنگاه هرکدام از مولفه‌ها با یک عدد صحیح بین 0 تا 255 نمایش داده می‌شوند.

همان‌گونه که پیشتر ذکر شد، پنج ویژگی اصلی در تعیین خطرپذیری مدارهای دیجیتال در برابر تروجان‌های سخت‌افزاری موثرند (فضاهای خالی، تراکم منابع مسیردهی و مسیرهای غیربحرانی، فعالیت سیگنالی گره‌های مدار، میزان مشاهده‌پذیری و مقدار کنترل‌پذیری گره‌های مدار). این ویژگی‌ها باید به صورت مناسبی به سه مولفه قرمز، سبز و آبی در تصاویر رنگی نگاشته شوند. باتوجه به ارتباط تنگاتنگ سه ویژگی فعالیت سیگنالی، مشاهده‌پذیری و کنترل‌پذیری گره‌های مدار باهم، این سه ویژگی ترکیب شده و با یک عدد ۸ بیتی به مؤلفه سبز هر پیکسل از تصویر رنگی نگاشته می‌شود. برای این کار، ابتدا مقدار CC0، CC1، CO و فعالیت سیگنال (act) هرکدام از گره‌های مدار محاسبه و نرمالیزه می‌شوند. برای نرمالیزه کردن هر مشخصه مقدار آن در واحد مربع موردنظر را بر ماکزیمم مقدار آن مشخصه در کل چیدمان فیزیکی تقسیم می‌شود تا یک عدد در بازه [۰،۱] حاصل شود. این مقادیر نرمالیزه شده با استفاده از رابطه (۱۰) با هم ترکیب می‌شوند و حاصل عملیات با استفاده از تبدیل اعداد ممیز ثابت[1] به یک عدد ۸ بیتی تبدیل می‌گردد. این مقدار نهایی به عنوان مؤلفه سبز رنگ پیکسل معادل از تصویر RGB ذخیره می‌شود.

$$RR_{norm} = \sqrt{(CC0+CC1+CO)^2 + act^2} \qquad (۱۰)$$

در رابطه (۱۰)، CC0، CC1، CO^2 و act^3 به ترتیب بیانگر میزان کنترل‌پذیری ترکیبی 0، کنترل‌پذیری ترکیبی 1، مشاهده‌پذیری ترکیبی و مقدار فعالیت سوئیچینگ گره‌های مدار می‌باشند.

با توجه به تعریف مقدار فضای خالی در هر مربع واحد که برابر نسبت مساحت خالی به کل مساحت مربع واحد می‌باشد، عدد مربوطه همواره در بازه ۰ تا ۱ بوده و بنابراین نیاز به نرمالیزه کردن، ندارد. نسبت به‌دست‌آمده با استفاده از تبدیل عدد اعشار به عدد ممیز ثابت، به یک عدد ۸ بیتی تبدیل می‌شود و به مولفه قرمز رنگ در پیکسل معادل نسبت داده می‌شود. به عنوان مثال اگر ۳/۴ مساحت یک مربع از سطح تراشه فضای خالی و مابقی با چیدمان فیزیکی گیت‌های منطقی پر شده باشد، مقدار عدد معادل ممیز ثابت برابر ۱۹۲ می‌باشد که به صورت عدد باینری 11000000 ذخیره خواهد شد.

مؤلفه آبی پیکسل‌ها به مقدار ویژگی تراکم منابع مسیردهی و مسیرهای غیربحرانی در هر مربع واحد، مربوط می‌شود. برای محاسبه عدد نرمالیزه معادل، ابتدا تعداد سیم‌هایی که برای مسیردهی سیگنال‌های مدار (از مسیرهای غیربحرانی) از مربع موردنظر عبور می‌کند، محاسبه می‌شود. سپس این عدد بر تعداد کل سیم‌هایی که قابلیت عبور از سطح مربع واحد موردنظر را دارند، تقسیم می‌گردد. عدد نرمالیزه حاصل با استفاده از روش تبدیل اعداد ممیز ثابت به

[1] Fixed point
[2] Combinational Observability
[3] Switching Activity

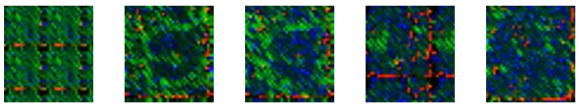
یک عدد باینری ۸ بیتی تبدیل می‌شود و به عنوان مولفه آبی پیکسل موردنظر در تصویر RGB ذخیره می‌گردد. چنین محاسبه‌ای از آن جهت منطقی می‌باشد که فرد طراح تروجان سخت‌افزاری می‌تواند از ظرفیت استفاده نشده برای سیم‌کشی در سطح مربع واحد برای ایجاد مسیرهای موردنیاز جهت اتصال تریگر به هسته تروجان سخت‌افزاری، استفاده کند.

بستر نرم‌افزاری توسعه یافته در این پژوهش قادر است که مدارهای دیجیتال را ازنظر میزان خطرپذیری به ده دسته با خطرپذیری‌های مختلف، کلاس‌بندی کند. چون آموزش شبکه عصبی عمیق نیازمند داده‌های برچسب دار است، بنابراین در مجموعه‌داده تولیدشده باید هر کدام از مدارهای پیاده‌سازی شده با برچسب کلاس‌های ده گانه برچسب‌گذاری شوند. برای تعیین این برچسب‌ها، انواع مختلفی تروجان سخت‌افزاری تولیدشده و امکان قراردادن آن‌ها در مدار پیاده‌سازی شده، بررسی شده است.

جهت محاسبه میزان سختی قراردادن هر تروجان در مدار پیاده‌سازی شده، مراحل ذیل طی می‌شود. مرحله اول شامل یافتن فضای خالی مناسب جهت قراردان هسته تروجان موردنظر است. درصورتی که چنین فضایی وجود نداشته باشد، فرآیند قراردادن تروجان شکست می‌خورد و میزان خطرپذیری مدار دیجیتال کاهش می‌یابد. درصورت یافتن جای مناسب، تعداد امکانات لازم برای تریگر کردن آن تروجان و وجود منابع مسیردهی لازم برای اتصال آن تریگر به هسته تروجان بررسی می‌شود. برای هر تریگر، امتیازی بر اساس میزان آشکارسازی و سختی استفاده از آن تریگر محاسبه می‌شود و درنهایت برای تمام تروجان‌های قابل پیاده‌سازی، امتیازات جمع شده و به عنوان امتیاز کلی آن مدار ثبت می‌گردد.

پس از محاسبه امتیازِ تمام ۱۰۰۰۰ مدار پیاده‌سازی شده، تصاویر براساس امتیازها به صورت نزولی مرتب‌سازی می‌شوند. بالاترین امتیاز مربوط به مدارهای با خطرپذیری بسیار زیاد و پایین‌ترین امتیاز مربوط به مدارها با خطرپذیری کم است. این بازه به ۱۰ کلاس مختلف طبقه‌بندی می‌شوند که در آن شماره کلاس بیشتر، نمایانگر خطرپذیری بیشتر است. درنهایت هرکدام از مدارهای پیاده‌سازی شده در مجموعه‌داده با یک برچسب که تعیین‌کننده کلاس خطرپذیری آن است، مشخص می‌شود.

در شکل ٤ چند نمونه از تصاویر مربوط به کلاس‌های مختلف از مجموعه‌داده تولیدشده نشان داده‌شده است.



شکل (٤): چند تصویر از مجموعه داده تولیدشده

۳-۲ شبکه‌های عصبی پیچشی

شبکه‌های عصبی کانولوشن، جزء پرکاربردترین راهکارهای یادگیری عمیق در زمینه‌های مختلف می‌باشند [۱۳]. از لحاظ معماری، این شبکه‌ها یک نوع شبکه عصبی Feed-forward می‌باشند که داده‌ها را معمولاً به صورت ماتریس‌های دوبعدی دریافت می‌کنند. بخش‌های اصلی شبکه عصبی کانولوشن شامل لایه ورودی، لایه پیچشی، لایه ادغام[1]، لایه مسطح کننده، لایه کاملاً متصل و لایه خروجی می‌باشد. یک نمونه از شبکه‌های عصبی کانولوشن در شکل ۵ نشان داده‌شده است.

وظیفه اصلی لایه پیچشی استخراج ویژگی‌های مختلف از ورودی شبکه است. بعضی از لایه‌های پیچشی (لایه‌های نزدیک به لایه ورودی) وظیفه استخراج ویژگی‌های سطح پایین مانند لبه‌ها، خطها و گوشه‌ها را برعهده دارند. لایه‌های بعدی برای استخراج ویژگی‌های پیچیده‌تر، با ترکیب ویژگی‌های سطح پایین استخراج‌شده در لایه‌های ابتدایی، به‌کار می‌روند. با وجود توانایی استخراج ویژگی‌ها توسط لایه‌های پیچشی، برای انجام تصمیم‌گیری صحیح در مورد کلاسِ تصویر ورودی توسط لایه کاملاً متصل، لازم است که تعداد ویژگی‌های استخراج‌شده به صورت هوشمندانه‌ای کاهش یابند. این امر با استفاده از لایه ادغام محقق می‌شود. لایه ادغام با نمونه‌برداری از ویژگی‌های استخراج‌شده به وسیله لایه پیچشی، به کاهش تعداد پارامترهای مؤثر در شبکه کمک می‌کند. پرکاربردترین نوع نمونه‌برداری در لایه ادغام استفاده از عملیات ماکزیمم (max pooling) و میانگین (average pooling) می‌باشد. در روش ادغام ماکزیمم، مقدار بیشینه از پیکسل‌هایی که توسط پنجره ادغام پوشش داده می‌شوند انتخاب می‌گردد و در روش ادغام میانگین مقدار متوسط پیکسل‌ها به عنوان خروجی درنظر گرفته می‌شود. در هردو روش یک پیکسل به نمایندگی از کل پیکسل‌های پنجره به خروجی لایه ادغام می‌رود که این امر منجر به کاهش تعداد ویژگی‌ها برای لایه بعدی می‌گردد.

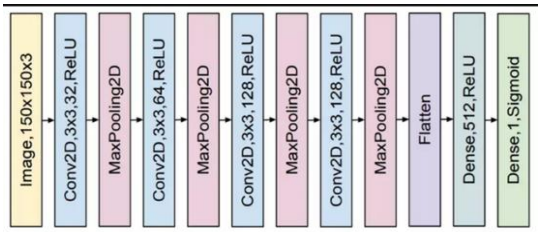
[1] Pooling

از آنجاییکه لایه تماماً متصل، یک بردار از داده‌ها را به‌صورت ورودی می‌پذیرد، لذا خروجی آخرین لایه پیچشی که به صورت ماتریس‌های دوبعدی می‌باشد، به‌وسیله لایه مسطح کننده به یک بردار تبدیل می‌شود. لایه تماماً متصل از شبکه عصبی پیچشی، ساختاری شبیه شبکه عصبی[1] MLPدارد که در آن خروجی نرون‌های موجود در هر لایه به ورودی تمام نرون‌های لایه بعد متصل می‌شوند. وظیفه اصلی این بخش از شبکه عصبی پیچشی، تشخیص کلاس بردار خروجی لایه مسطح کننده می‌باشد. لازم به‌ذکر است که لایه آخر در ساختار بخش تماماً متصل، از تابع softmax برای تشخیص کلاس داده ورودی موردنظر استفاده می‌کند (رابطه ۱۱).

$$softmax(x_i) = \frac{e^{-\beta x_i}}{\sum_{j=1}^{K} e^{-\beta x_j}} \qquad (11)$$

در رابطه (۱۱)، فرض بر این است که بردار ورودی تابع softmax شامل K عنصر می‌باشد ($vector=(x_1, x_2,\ldots, x_K)$) و مشخصه)با توجه به شرایط مساله تعیین می‌شود.

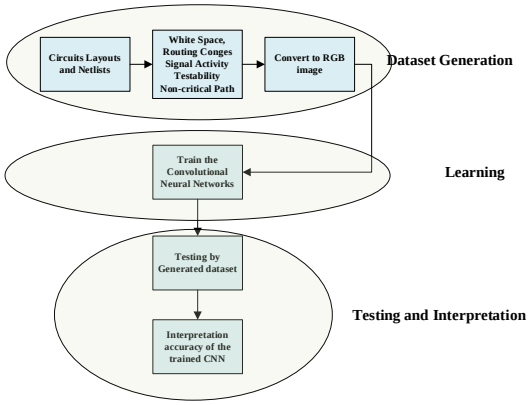
آموزش شبکه‌های پیچشی که شامل تنظیم پارامترهای مختلف شبکه (وزن‌ها، مقادیر بایاس و ...) است به صورت آموزش با نظارت است. در این نوع یادگیری از یک مجموعه داده که در آن برروی هرکدام از داده‌ها برچسب مناسب قراردادهشده است، استفاده می‌شود. در ابتدای مرحله یادگیری شبکه، مقادیر پارامترها به صورت تصادفی انتخاب می‌شوند. سپس داده‌های موجود در مجموعه‌داده به صورت تصادفی به شبکه اعمال می‌شوند. برای هرکدام از داده‌ها، مقدار خروجی شبکه استخراج می‌شود و سپس با توجه به مقدار خطای تولیدشده (بر اساس یک تابع هزینه)، در مرحله پس انتشار[2]، مقدار پارمترهای قابل تنظیم شبکه اصلاح می‌گردند. پس از تکرار چندین باره الگوریتم یادگیری، شبکه عصبی پیچشی در بهترین وضعیت خود قرار می‌گیرد. درنهایت با استفاده از داده‌های آزمون مقدار دقت شبکه عصبی سنجیده می‌شود.



شکل (۵): نمونه‌ای از معماری شبکه عصبی کانولوشن

۳-۳ روند پیشنهادی کلاس‌بندی مدارهای دیجیتال با استفاده از شبکه عصبی کانولوشن

در شکل (٦) روند پیشنهادی این مقاله نشان داده‌شده است. در مرحله اول، داده‌های مورد نیاز (مجموعه داده) برای اجرای عملیات یادگیری شبکه عصبی پیچشی و ارزیابی صحت نتایج تولید می‌شود. توضیحات مربوط به این مرحله در بخش (۳-۱) ذکر گردیده است.



شکل (٦): جزئیات گام‌های مختلف در روند پیشنهادی

در مرحله دوم، معماری شبکه عصبی پیچشی مورد نظر انتخاب می‌شود و براساس مجموعه داده تولیدشده آموزش می‌بینند. در مرحله آموزش ٪۷۰ داده‌های کل به صورت تصادفی انتخاب می‌شوند که در میان این داده‌ها نیز، ٪۲۰ برای اعتبارسنجی کنارگذاشته می‌شوند.

در مرحله سوم، با استفاده از ٪۳۰ داده انتخاب شده به عنوان داده آزمون، میزان دقت شبکه آموزش یافته با استفاده از معیارهای مختلف سنجیده و تحلیل می‌شود.

مساله بسیار مهم در فرآیند سه مرحله‌ای پیشنهادی، نحوه تعیین مقدار بهینه ابرمشخصه‌های[3] می‌باشد. یکی از راهکارهای پرکاربرد در این زمینه، استفاده از روش جستجوی حریصانه است. در این روش ابتدا یک لیست از ابرمشخصه‌های مورد نیاز شبکه (نرخ یادگیری، اندازه دسته، تعداد epoch و ...) تهیه می‌شود و گستره مقدار هرکدام از آن‌ها، به ۲۵ مقدار مختلف تقسیم‌بندی می‌شود. در ابتدا مقادیر اولیه هرکدام از ابرمشخصه‌ها تعیین می‌شوند، سپس مرحله آموزش شبکه با استفاده از این ابرمشخصه‌ها تکمیل می‌شود و عملکرد شبکه ارزیابی می‌شود. در مرحله بعد، مقدار جدیدی از ابرمشخصه‌ها انتخاب می‌شود و شبکه بازآموزی می‌شود واگر شبکه با ابرمشخصه‌های جدید عملکرد بهتری نسبت به بهترین حالت قبلی آن‌ها داشته باشد، ابرمشخصه‌های جدید به عنوان حالت بهینه انتخاب

[1] Multilayer Perceptron
[2] Back propagation
[3] Hyper-parameters

می‌شوند و در غیر این‌صورت مقدار بهینه قبلی حفظ می‌شود.. هرچند این روش نسبت به دیگر الگوریتم‌ها (جستجوی اتفاقی و بهینه‌سازی Bayesian) بیشتر طول می‌کشد، لیکن ابرمشخصه‌های انتخاب شده عملکرد بسیار خوبی دارند.

ذکر این نکته ضروری به نظر می‌رسد که در مدلسازی با استفاده از شبکه‌های عصبی پیچشی، باید مساله انطباق بیش از حد[1] به طریق مناسبی حل شود. برای اینکار روش‌های مختلفی همچون استفاده از روش‌های انتظام[2]، توقف زودهنگام[3]، حذف کردن[4] و غیره ارائه شده است. L2-Regulization و ۲۰٪ حذف کردن به همراه توقف زودهنگام راهکارهای جلوگیری از انطباق بیش از حد مورد استفاده در مرحله آموزش در شبکه عصبی پیچشی پیشنهادی این پژوهش می‌باشند.

٤- شبیه‌سازی‌ها و نتایج

در این بخش به بررسی دقت روش پیشنهادی در کلاس‌بندی خطرپذیری تراشه‌های دیجیتال در برابر تروجان‌های سخت‌افزاری پرداخته می‌شود. شبیه‌سازی‌ها در محیط MATLAB و Python (کتابخانه Tensorflow مربوط به یادگیری عمیق) برروی سیستم کامپیوتری با پردازنده نسل ٦ از Core-i7 با حافظه RAM، ۸ گیگابایت و فرکانس ۲/۵ گیگاهرتز و کارت گرافیکی GeForce 940MX انجام‌شده‌اند. در مرحله یادگیری شبکه‌های عصبی عمیق، از بهینه‌ساز GCD، نرخ یادگیری ۰/۰۰۲، تعداد epoch برابر ۱۰۰ و تابع هزینه categorical cross-entropy استفاده‌شده است. همان گونه که در بخش پیش ذکر شد، مقادیر بهینه ابرمشخصه‌ها با استفاده از روش جستجوی حریصانه برروی شبکه کامل هایپرپارمترها استخراج‌شده‌اند.

این پژوهش جهت اندازه‌گیری میزان صحت روند پیشنهادی از مفاهیم ذیل بهره می‌برد:

TP[5]: مواردی از مجموعه‌داده که برای آن‌ها، کلاس پیش‌بینی شده توسط شبکه عصبی عمیق با کلاس واقعی تصویر یکسان است.

TN[6]: مواردی از مجموعه‌داده که برای آن‌ها، شبکه عصبی عمیق به درستی تشخیص می‌دهد تصویر مورد نظر متعلق به کلاس تحت بررسی نیست.

FP[7]: مواردی از مجموعه‌داده که برای آن‌ها، شبکه عصبی به اشتباه تشخیص می‌دهد تصویر موردنظر متعلق به کلاس تحت بررسی است.

FN[8]: مواردی از مجموعه‌داده که برای آن‌ها، شبکه عصبی به اشتباه تشخیص می‌دهد تصویر موردنظر متعلق به کلاس تحت بررسی نیست.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (۱۲)$$

$$TPR = TP / (TP+FN) \quad (۱۳)$$

$$TNR = TN/(TN+FP) \quad (۱٤)$$

با توجه به تعاریف بالا و روابط (۱۲-۱٤) سه معیار زیر برای محاسبه کارایی شبکه‌های عصبی آموزش یافته مطابق روند پیشنهادی، مورد استفاده قرار می‌گیرند.

۱- صحت: طبق رابطه (۱۲)، نسبت تصاویری از مجموعه‌داده که به صورت صحیح توسط شبکه عصبی عمیق کلاس‌بندی می‌شود به کل تصاویر.

۲- TPR[9]: طبق رابطه (۱۳)، نسبت تصاویری که در یک کلاس قراردارند و به درستی کلاس‌بندی شده‌اند به کل تصاویر موجود در آن کلاس.

۳- TNR[10]: طبق رابطه (۱٤)، نسبت تعداد تصاویری که شبکه عصبی عمیق به‌درستی تشخیص داده است که متعلق به کلاس تحت بررسی نیست به کل تصاویری که در آن کلاس قرار نمی‌گیرند.

در جدول (۲)، ساختار شبکه عصبی پیچشی و مقادیر ابرمشخصه‌ها که پس از اعمال الگوریتم جستجوی حریصانه به‌دست‌آمده است، گزارش‌شده است.

به دلیل اینکه روش‌های پیشینِ برمبنای یادگیری ماشین، تنها برای ***تشخیص*** وجود تروجان‌های سخت‌افزاری بعد ازتولید تراشه توسعه پیدا کرده‌اند [۱٤-۱۷] و به ارزیابی میزان خطرپذیری تراشه‌ها در برابر تروجان‌های سخت‌افزاری در مرحله طراحی نپرداخته‌اند، استخراج نتایج برای طبقه‌بندی کننده‌های[11] معمولی برروی مجموعه‌داده این پژوهش، به صورت مستقل انجام‌شده است. در جدول (۳) مقایسه میزان دقت به‌دست‌آمده توسط شبکه عصبی پیچشی با سایر روش‌های طبقه‌بندی کننده خطی مشهور گزارش‌شده است. لازم به ذکر است که نتایج مربوط به طبقه‌بندی کننده‌های معمولی (طبقه‌بندی کننده‌های مرکب[12]، ماشین

[1] Overfitting
[2] Regularization
[3] Early Stopping
[4] Dropout
[5] True Positive
[6] True Negative
[7] False Positive
[8] False Negative
[9] True Positive Rate
[10] True Negative Rate
[11] Classifiers
[12] Ensemble Classifier

بردار پشتیبان[1]، درخت[2]، k –همسایه نزدیک‌تر[3] با آموزش آن‌ها در Toolbox مربوطه از نرم‌افزار MATLAB صورت گرفته است. برای آموزش این طبقه‌بندی کننده‌ها از مجموعه داده تولیدشده در این پژوهش استفاده‌شده است و مقدار ابرمشخصه‌ها با استفاده از روش جستجوی حریصانه موجود در Toolbox استخراج‌شده است.

همان‌گونه که ملاحظه می‌شود، میزان دقت شبکه پیشنهادی (بالای ۹۲٪) نسبت به سایر روش‌ها بسیار بهتر است. یکی از دلایل این عملکرد بهتر را می‌توان در توانایی فوق‌العاده لایه‌های پیچشی شبکه عصبی پیشنهادی ما برای استخراج ویژگی‌های اساسی مؤثر در میزان خطرپذیری تراشه‌های دیجیتال دانست. این لایه‌ها قادرند تعداد زیادی ویژگی مختلف را در تصویر ورودی پردازش کنند و درنهایت آن‌ها را به‌گونه‌ای به لایه‌های تماماً متصل ارائه دهند که بیشترین تفاوت را در کلاس‌های مختلف داشته باشند. ازسوی دیگر، الگوریتم فراگیر و پیچیده شبکه‌های عصبی پیچشی امکان تحلیل ویژگی‌های اصلی را در تصاویر مجموعه داده فراهم می‌کند، که این امکان با استفاده از الگوریتم‌های یادگیری ساده‌تر سایر طبقه‌بندی‌کننده‌ها وجود ندارد. همان‌گونه که پیشتر ذکرشد، ویژگی‌های تعیین‌کننده میزان خطرپذیری تراشه‌های دیجیتال، دارای رفتار غیرخطی در نواحی مختلف سطح تراشه می‌باشند و درنتیجه ترکیب اثرات این ویژگی‌ها و اتخاذ تصمیم در مورد میزان خطرپذیری کل تراشه فرآیندی بسیار پیچیده است که نیازمند استفاده از مدلی غیرخطی با توانایی یادگیری بالا دارد.

جدول (۲) مقادیر بهینه ابرمشخصه‌ها

ابرمشخصه	مقدار بهینه
تعداد لایه‌های پیچشی	5
تعداد لایه‌های ادغام	5
تعداد لایه‌های تماماً متصل	3
مومنتوم	0.9
نرخ یادگیری اولیه	0.1
ضریب کاهش نرخ یادگیری	0.09
نرخ تضعیف میانگین متحرک	0.95
تعداد دورها در هر تکرار	200
تضعیف وزن‌ها	0.0005
اندازه دسته	128

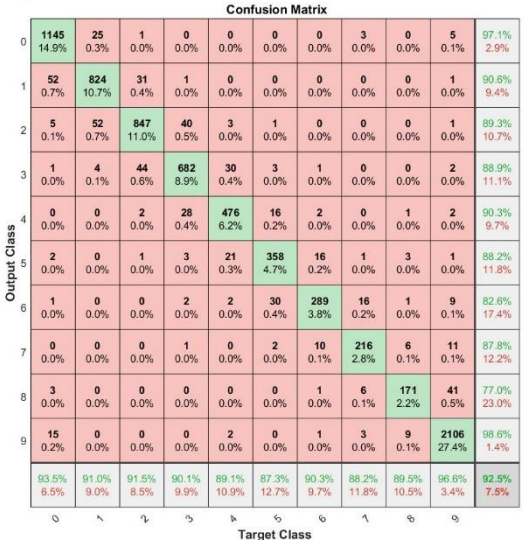
جدول (۳) میزان دقت طبقه‌بندی در روش‌های مختلف

Method	Accuracy%
Ensemble	75.7
Tree	67.5
SVM	72.4
KNN	68.2
[9]	62.4
[10]	57.8
[11]	54.8
Proposed CNN	92.6

مقایسه معیار دقت نتایج شبکه عصبی کانولوشن و بهترین الگوریتم‌های کلاس‌بندی خطی مورداستفاده در شبیه‌سازی‌ها (طبقه‌بندی کننده مرکب) بر مبنای ماتریس Confusion در شکل (۷) نشان داده‌شده است. برای استخراج این ماتریس‌ها از یک مجموعه ۳۰۰۰ تایی از تصاویر مربوط به کلاس‌های ده‌گانه موجود در مجموعه داده که به صورت اتفاقی انتخاب‌شده‌اند، استفاده‌شده‌است. در ماتریس Confusion، هر سطر بیانگر کلاس تشخیص داده‌شده توسط طبقه‌بندی کننده و هر ستون بیانگر کلاسی است که داده مورد آزمایش واقعاً به آن تعلق دارد. برای یک طبقه‌بندی‌کننده ایده‌آل مقدار خانه‌هایی که روی قطر اصلی قرارندارند برابر 0 است. مقادیر درصدهای داخل هرخانه به‌جز سطر آخر و همچنین ستون سمت راست، از تقسیم عدد مربوط به آن خانه برمجموع تمام اعداد داخل جدول (کل نمونه‌های استفاده‌شده) به دست می‌آید. اعداد درصد سبزرنگ مربوط به خانه‌های ستون سمت راست از تقسیم عدد موجود در مربع سبزرنگ سطر مربوطه بر تمام اعداد موجود درآن سطر به‌دست می‌آید و عدد درصد قرمزرنگ از تفریق درصد سبزرنگ از عدد ۱۰۰ به‌دست می‌آید. رویکرد مشابهی برای محاسبه درصدهای سبز و قرمزرنگ در خانه‌های سطر آخر اتخاذشده است. بنابراین، اعداد سبزرنگ ستون سمت راست بیانگر آن هستند که طبقه‌بندی کننده در مورد داده‌هایی که تشخیص داده است کلاس آن‌ها در ردیف خاصی قرار دارد، چه میزان صحیح عمل کرده‌است و همچنین درصد سبزرنگ هر ستون در سطر آخر بیان می‌کند که طبقه‌بندی کننده در مورد تشخیص کلاس داده‌های هرکدام از کلاس‌های ده‌گانه به‌درستی عمل کرده است. به‌طور مثال، در مورد داده‌هایی که متعلق به کلاس شماره ٤ هستند، شبکه‌عصبی پیشنهادی دارای دقت ۸۹/۱٪ و طبقه‌بندی مرکب دارای دقت ۵۷/۷٪ می‌باشد.

ازمقایسه جداول Confusion این نتیجه حاصل می‌شود که شبکه عصبی کانولوشن پیشنهادی، نسبت به دیگر الگوریتم‌ها از دقت بالاتری برای

[1] Support-Vector Machine (SVM)
[2] Tree
[3] K-nearest neighbors (KNN)



الف

ب

شکل (۷): الف) ماتریس confusion شبکه عصبی پیچشی ب) ماتریس confusion طبقه‌بندی کننده مرکب

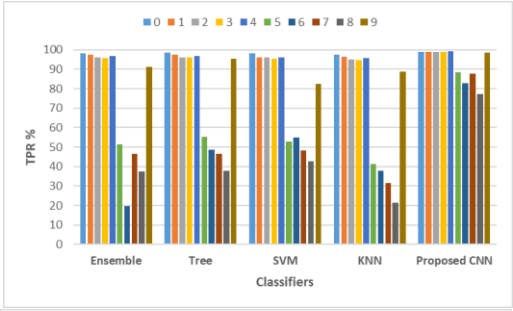
تشخیص کلاس مربوط به هر مدار و همچنین دقت بالاتری در تشخیص عدم تعلق یک مدار به یک کلاس خاص برخوردار است.

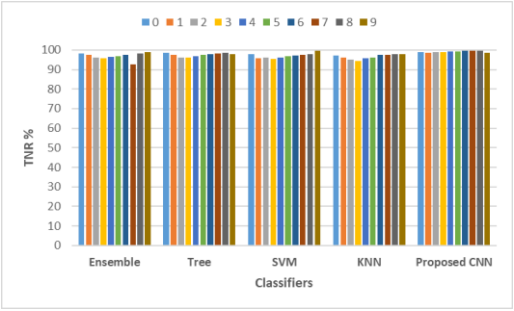
در شبیه‌سازی دیگری که نتایج آن در شکل (۸) نشان داده‌شده است، مقادیر TPR و TNR مربوط به شبکه عصبی کانولوشن پیشنهادی و دیگر الگوریتم‌های طبقه‌بندی کننده، گزارش‌شده است. همان‌گونه که پیشتر ذکرشد، TPR بیانگر میزان توانایی یک طبقه‌بندی کننده در تشخیص صحیح داده‌هایی است که به یک کلاس خاص تعلق دارند. این معیار در شبکه‌عصبی پیچشی پیشنهادی در همه کلاس‌ها نسبت به سایر طبقه‌بندی کننده‌ها برتری محسوس دارد و این برتری مخصوصاً در کلاس‌های بالاتر (که میزان خطرپذیری بیشتری نسبت به تروجان‌های سخت‌افزاری دارند) به‌خوبی قابل‌مشاهده است. به‌عبارت‌دیگر، در سایر طبقه‌بندی کننده‌ها به‌احتمال بسیار بیشتر ممکن است یک مدار دیجیتال که میزان خطرپذیری بالایی دارد به‌عنوان مداری تشخیص داده شود که دارای خطرپذیری کم و یا متوسط است و این امر منجر به خطرپذیری بیشتر در قراردادن تروجان‌های سخت‌افزاری در این تراشه‌ها می‌شود.

درنهایت، برای مقایسه باکارهای پیشین نتایج صحت شبکه عصبی پیچشی پیشنهادی و الگوریتم‌های کلاس‌بندی با سه روش مراجع [۹]، [۱۰] و [۱۱] در جدول (۳) مقایسه شده‌اند. روش‌های مذکور بر اساس یادگیری ماشین نیستند و نتایج را برای مجموعه مدارهای دیگر ارائه کرده‌اند. لذا جهت مقایسه صحیح با روش‌های مبتنی بر یادگیری ماشین، نتایج آن‌ها (با کدنویسی روش‌های مذکور) برای مجموعه داده این پژوهش، بازتولیدشده‌است. همان‌گونه که در بخش (۳) ذکر گردید، این روش‌ها هرکدام تعدادی از مشخصه‌های اصلی دخیل در میزان خطرپذیری را در نظر نگرفته‌اند و یا مدل‌سازی آن‌ها از این مشخصه‌ها جامع و کامل نمی‌باشد. از نتایج جدول (۳) نیز می‌توان دریافت که اختلاف زیادی در میزان دقت این روش‌ها با روش‌های مبتنی بر الگوریتم‌های هوشمند وجود دارد. دلیل اصلی این امر را می‌توان در توانایی فوق العاده شبکه‌های عصبی عمیق پیچشی در استخراج ویژگی‌های مؤثر در تعیین میزان خطرپذیری تراشه‌های دیجیتال دانست. این ویژگی‌ها دارای روابط پیچیده با یکدیگر هستند که امکان مدلسازی جامع و دقیق آن‌ها هم با استفاده از روش‌های تحلیلی (closed form) و هم با استفاده از طبقه‌بندی کننده‌های خطی معمولی غیر ممکن است.

مقایسه نتایج مربوط به سه رهیافت مراجع [۹]، [۱۰] و [۱۱] نشان می‌دهد که عملکرد مرجع [۹] از دو روش دیگر بهتر است، که دلیل اصلی آن ارائه مدلی نسبتاً دقیق‌تر از فضاهای خالی چیدمان فیزیکی در این روش می‌باشد. لیکن عدم مدلسازی دقیق مسیردهی تریگر به هسته تروجان و عدم شناخت دقیق فرآیند تریگر، دقت آن را تا ٦٢٪ کاهش داده است. از طرفی دیگر، درنظر گرفتن معیارهای آزمون‌پذیری (کنترل‌پذیری و مشاهده‌پذیری) در رهیافت مرجع [۱۰] سبب شده است که این روش دقت بالاتری نسبت به رهیافت مرجع [۱۱] داشته باشد. اشکالات دیگر

روش مرجع [۱۱] عدم مدلسازی مناسب سیگنال‌های تریگر و همچنین غیرقابل اعمال آن برای مدارهای ترکیبی می‌باشد.





شکل (۸): مقایسه TPR و TNR برای طبقه‌بندی‌کننده‌های مختلف

۵- نتیجه‌گیری

در این مقاله، به مدل‌سازی میزان خطرپذیری تراشه‌های دیجیتال در برابر حملات امنیتی ناشی از تروجان‌های سخت‌افزاری پرداخته شده است. در ابتدا مجموعه‌داده مناسبی از پیاده سازی‌های متنوع تراشه‌های دیجتال ارائه شده و بر اساس روش قراردادن تروجان‌های سخت‌افزاری میزان خطرپذیری آن‌ها کلاس‌بندی شده‌اند. سپس مجموعه داده تولیدشده جهت آموزش شبکه عصبی کانولوشن مورد استفاده قرارگرفته و میزان توانایی شبکه پیشنهادی در استخراج ویژگی‌های مختلف، منجر به کلاس‌بندی با دقت ۹۲٪ شده است. مقایسه نتایج دقت کلاس‌بندی با سایر طبقه‌بندی کننده‌ها و همچنین روش‌های پیشین میزان بهبود ۱۷٪ را نشان می‌دهد. از مدل بدست آمده می‌توان در مراحل اولیه طراحی تراشه‌های دیجیتال برای استخراج میزان خطرپذیری مدارها و امکان به کارگیری رهیافت‌هایی برای ارتقاء امنیت تراشه‌ها بهره برد. با توجه به دقت بالای شبکه عصبی کانولوشن پیشنهادی در عین کم بودن سربار محاسباتی و حافظه، این امکان وجود دارد که از رهیافت پیشنهادی برای ارزیابی میزان خطرپذیری تراشه‌های دیجیتال در مرحله طراحی، در نرم‌افزارهای مربوط به طراحی با کمک کامپیوتر استفاده شود.

References

[1] Bao, C., Forte, D., & Srivastava, A. (2015). On reverse engineering-based hardware Trojan detection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35(1), 49-57.

[2] Bhunia, S., Abramovici, M., Agrawal, D., Bradley, P., Hsiao, M.S., Plusquellic, J. and Tehranipoor, M., 2013. Protection against hardware trojan attacks: Towards a comprehensive solution. IEEE Design & Test, 30(3), pp.6-17.

[3] He, J., Guo, X., Ma, H., Liu, Y., Zhao, Y. and Jin, Y., 2020, July. Runtime trust evaluation and hardware trojan detection using on-chip em sensors. In 2020 57th ACM/IEEE Design Automation Conference (DAC) (pp. 1-6). IEEE.

[4] Cozzi, M., Galliere, J.M. and Maurine, P., 2018, August. Exploiting Phase Information in Thermal Scans for Stealthy Trojan Detection. In 2018 21st Euromicro Conference on Digital System Design (DSD) (pp. 573-576). IEEE.

[5] Koushanfar, F. and Mirhoseini, A., 2010. A unified framework for multimodal submodular integrated circuits trojan detection. IEEE Transactions on Information Forensics and Security, 6(1), pp.162-174.

[6] Chen, K., Arias, O., Guo, X., Deng, Q., & Jin, Y. (2022). IP-Tag: Tag-Based Runtime 3PIP Hardware Trojan Detection in SoC Platforms. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42(1), 68–81

[7] Q. Shi, N. Vashistha, H. Lu, H. Shen, B. Tehranipoor, D. L. Woodard, and N. Asadizanjani, “Golden gates: A new hybrid approach for rapid hardware trojan detection using testing and imaging,” in 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). IEEE, 2019, pp. 61–71.

[8] Rahimifar, M. M., & Jahanirad, H. (2020). Employing Image Processing Techniques for Hardware Trojans Detection. In In 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 187–192). IEEE.

[9] Trippel, T., Shin, K. G., Bush, K. B., & Hicks, M. (2020). ICAS: An extensible framework for estimating the susceptibility of ic layouts to additive trojans. In In 2020 IEEE Symposium on Security and Privacy (SP) (pp. 1742–1759). IEEE.

on Computer-Aided Design of Integrated Circuits and Systems 37, no. 7 (2017): 1370-1383.

[15] Kok, Chee Hoo, Chia Yee Ooi, Michiko Inoue, Mehrdad Moghbel, Sreedharan Baskara Dass, Hau Sim Choo, Nordinah Ismail, and Fawnizu Azmadi Hussin. "Net classification based on testability and netlist structural features for hardware trojan detection." In 2019 IEEE 28th Asian Test Symposium (ATS), pp. 105-1055. IEEE, 2019.

[16] Hasegawa, Kento, Masao Yanagisawa, and Nozomu Togawa. "Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier." In 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-4. IEEE, 2017.

[17] [17] Hasegawa, Kento, Youhua Shi, and Nozomu Togawa. "Hardware trojan detection utilizing machine learning approaches." In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 1891-1896. IEEE, 2018.

[10] Salmani, H., & Tehranipoor, M. M. (2016). Vulnerability analysis of a circuit layout to hardware Trojan insertion. IEEE Transactions on Information Forensics and Security, 11 (6), 1214–1225.

[11] Bakhshizadeh, M., & Jahanian, A. (2014). Trojan Vulnerability Map: An efficient metric for modeling and improving the security level of hardware. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 97(11), 2218–2226.

[12] Zamiri Azar, Kimia, Hadi Mardani Kamali, Farimah Farahmandi, and Mark Tehranipoor. "Basics of VLSI Testing and Debug." In Understanding Logic Locking, pp. 25-46. Cham: Springer International Publishing, 2023.

[13] Cong, Shuang, and Yang Zhou. "A review of convolutional neural network architectures and their optimizations." Artificial Intelligence Review 56, no. 3 (2023): 1905-1969.

[14] Chen, Xiaoming, Qiaoyi Liu, Song Yao, Jia Wang, Qiang Xu, Yu Wang, Yongpan Liu, and Huazhong Yang. "Hardware trojan detection in third-party digital intellectual property cores by multilevel feature analysis." IEEE Transactions

Journal of Applied And Basic Machine Intelligence Research (ABMIR)

Applied and Basic Machine Intelligence Research

Vol. 2, No. 1, Spring & Summer 2024, pp. 51-66

A convolutional Neural Networks-based Approach for Vulnerability Classification of Integrated Circuits against Hardware Trojans

Hadi Jahanirad*

Department of Electronics and Communication Engineering, University of Kurdistan, Sanandaj, Iran

Article Information

Original Research Paper

Received:
2024 March 7

Accepted:
2024 August 27

Keywords:
Digital integrated circuits, Hardware Trojans, Convolutional neural networks, Classification, Machine learning.

Corresponding Author*:
h.jahanirad@uok.ac.ir

Abstract

The vulnerability of digital integrated circuits against the Hardware Trojans (HT) has increased in recent decades due to the implementation of more complex systems on them. HTs could become a source of errors or apply to steel important information embedded in the implemented circuits. So, analyzing the vulnerability of digital integrated circuits in the early stages of production is of great merit. In this paper, a novel vulnerability classification method is introduced based on the deep convolutional neural networks (CNN) wherein five major effective features of vulnerability assessment are utilized (white space distribution, unutilized routing resources, signal activity of circuit nodes, delay of the circuit paths and, controllability of circuit nodes). In the proposed framework, first of all, a dataset containing 10000 images is generated using various digital circuit implementations. Then, a deep CNN is trained using the generated dataset meanwhile the most appropriate CNN's hyperparameters are achieved using a greedy optimization method. The simulation results reveal 92% accuracy of vulnerability classification which shows a 17% improvement in comparison with the best linear classifier and analytical methods.

: 10.22034/ABMIR.2024.21351.1049

E-ISSN: 2821-2037