
بسم الله الرحمن الرحيم

چکیده

با توسعه صنعت نرم‌افزار و افزایش کیفیت و قابلیت اتکای محصولات نرم‌افزاری، آزمون نرم‌افزار موردتوجه مهندسین قرار گرفت. اما ازآنجایی‌که آزمون تمام ورودی‌ها حتی برای یک تابع ساده مشکل است و آزمون تمام واحدهای نرم‌افزاری نیز اغلب زمان‌بر و پرهزینه است، معمولاً از انجام آن صرف‌نظر می‌شود. به‌این‌ترتیب برای کاهش هزینه کلی آزمون و کشف خطاهای احتمالی پیش از وقوع آن‌ها، روش پیش‌بینی خطای نرم‌افزار مورداستفاده قرار می‌گیرد. یکی از چالش‌های موجود در زمینه پیش‌بینی خطای نرم‌افزار این است که مدل‌های پیش‌بینی قبلی، مدل‌های قدیمی‌ای بودند یا مجموعه داده‌هایی را به کار می‌گرفتند که روی پروژه‌های کوچک، محدود و قدیمی بودند؛ که به دلیل تغییر روند توسعه نرم‌افزارها، این مدل‌ها کارایی چندانی را برای نرم‌افزارهای امروزی ارائه نمی‌دهند. در این تحقیق برای کمک به فرایند پیش‌بینی خطا در مقابل روش‌های قدیمی، مجموعه‌داده جدیدی به نام باگ‌هانتر مورداستفاده قرارگرفته است. این مجموعه‌داده ارزشمند و نسبتاً بزرگ، متشکل از ۱۵ برنامه مرسوم و منبع باز جاوا است که امروزه به‌طور گسترده‌ای توسط مهندسین نرم‌افزار استفاده می‌شوند. تاکنون تکنیک‌های گوناگون یادگیری ماشین نظیر بیز ساده، رگرسیون لجستیک، جنگل تصادفی و غیره جهت کشف خطاهای نرم‌افزاری برای این مجموعه‌داده مورداستفاده قرارگرفته است. چالش دیگری که وجود دارد این است که دقت مدل‌های پیش‌بینی موجود، برای این مجموعه داده پایین است. این تحقیق باهدف افزایش عملکرد مدل پیش‌بینی خطا، تکنیک یادگیری عمیق و تکنیک ترکیبی را به کار می‌گیرد و با ترکیب شبکه عصبی و جنگل تصادفی دقت مدل پیش‌بینی را افزایش می‌دهد. شبکه عصبی مورداستفاده در این پروژه پرسپترون چندلایه و طبقه‌بندی کننده مورداستفاده در آن، مدل ترتیبی به همراه چندین لایه متراکم و حذف تصادفی است. روش ترکیبی پیشنهادی نیز از ترکیب یک شبکه عصبی و جنگل تصادفی تشکیل‌شده است. برای ایجاد مدل‌های پیش‌بینی قدرتمند و جلوگیری از جانب‌دار شدن مدل‌ها، تکنیک نمونه‌برداری تصادفی رو به بالا مورداستفاده قرارگرفته است. درنهایت روش پیشنهادی بر روی مجموعه داده باگ‌هانتر اعمال شد و نتایج بهبود روند پیش‌بینی خطا را به‌خصوص برای تکنیک ترکیبی پیشنهادی نشان می‌دهد. به‌طوری‌که مقدار معیار امتیاز اف-۱ به‌دست‌آمده توسط تکنیک شبکه عصبی پیشنهادی ۸۱٪/۷۴ است و توسط تکنیک ترکیبی پیشنهادی ۸۴/۸۱٪ است که نسبت به بهترین روش قبل، به ترتیب افزایش ۸٪ و ۱۱٪ داشته است.

کلمات کلیدی: پیش‌بینی خطای نرم‌افزار، یادگیری عمیق، یادگیری ماشین، یادگیری ترکیبی، مجموعه‌داده BugHunter، عدم تعادل کلاس

فهرست مطالب

فهرست جداول

فهرست اشکال

فصل اول: مقدمه

۱-۱ پیش گفتار

پیشرفت‌های اخیر تکنولوژی، رشد شدید صنعت نرم‌افزار را به همراه داشت. با افزایش تقاضای نرم‌افزار برای کاربردهای مختلف، به‌مرورزمان پیچیدگی و اندازه نرم‌افزارها گسترش یافت. ازاین‌رو توسعه نرم‌افزار باکیفیت، قابلیت اتکا و امنیت در دنیای مدرن جایگاه ویژه‌ای یافت. در این حیطه مهندسین نرم‌افزار نقش اساسی را ایفا می‌کنند.

وجود خطاها[۱] یکی از اصلی‌ترین مسائلی است که کیفیت و قابلیت اتکای نرم‌افزار را به چالش می‌کشاند. برنامه‌ها و کاربردهای نرم‌افزاری مختلف توسط برنامه‌نویسان و با استفاده از کدها ایجاد می‌شوند تا نیازمندی‌های کاربر نهایی از محصول موردنظر را برآورده سازند. حتی با توسعه‌ی دقیق محصولات و برنامه‌های نرم‌افزار، ممکن است کدها دارای خطاهای پیش‌بینی‌نشده باشند که از دید برنامه‌نویسان پنهان بمانند؛ بنابراین خروجی‌ها و عملکردهای مورد انتظار کاربر با خروجی‌ها و عملکردهای واقعی سیستم نرم‌افزاری در تناقض خواهد بود و کیفیت نرم‌افزار سلب می‌شود. ازاین‌رو پیش‌بینی و برطرف کردن خطاهای احتمالی و تضمین کیفیت نرم‌افزار قبل از عرضه به مشتری امری حیاتی در حوزه‌ی مهندسی نرم‌افزار است. خطای نرم‌افزار به یک مرحله، فرایند، یا تعریف نادرست داده در برنامه گفته می‌شود که باعث می‌شود برنامه به روشی پیش‌بینی‌نشده انجام شود و نتایج مطلوب و مورد انتظار آن فراهم نشود. خطاهای نرم‌افزار ممکن است در هر یک از مراحل توسعه به وجود آید و سیستم را با شکست مواجه کند. در ادامه چند تعریف پراستفاده در این مبحث ارائه‌شده است.

نقص:[۲] زمانی اتفاق می‌افتد که وظایف و نیازمندی‌های سیستم ارائه‌شده دارای کاستی و انحراف باشد و به‌درستی برآورده نشود. می‌توان گفت نقص نرم‌افزار نتیجه‌ی عدم انطباق بین نتایج مورد انتظار و واقعی سیستم‌های نرم‌افزاری است. بسته به‌اندازه و پیچیدگی نرم‌افزار، یافتن و برطرف کردن این نقص‌ها

[۱] Bugs
[۲] Defect

می‌تواند به‌آسانی و با سرعت بالایی انجام شود یا بسیار سخت و زمان‌بر باشد. به‌طورکلی این نقص‌ها توسط برنامه‌نویسانی که فرآیندهای طراحی‌شده را پیاده‌سازی می‌کنند، ایجاد می‌شوند.

خطا:[۱] یک مشکل ایستا در برنامه، یا خطای واضح و آشکاری است که در محصولات نرم‌افزاری رخداده است و مهندس آزمون باید آن را پیدا کند. خطاهای نرم‌افزاری توسط استفاده نادرست برنامه‌نویس از عملگرها یا عملوندهای برنامه به وجود می‌آیند و توسط ناظر کشف می‌شوند.

شکست:[۲] مشکل یا خطای موجود در برنامه منجر به یک شکست می‌شود. این شکست ممکن است منجر به توقف عملکرد مورد انتظار سیستم نرم‌افزاری شود، یا نتایج نادرست را برای ورودی‌های تعیین‌شده توسط مشتریان محصول نرم‌افزاری به دنبال داشته باشد. به‌عبارت‌دیگر با اجراشدن کد نرم‌افزاری دارای خطا، شکست اتفاق می‌افتد و ممکن است عملکرد سیستم نرم‌افزاری را مختل کند.

اشتباه:[۳] این امر اشتباه انسان است که نتایج نادرست را به دنبال دارد و منجر به خطا می‌شود.

برنامه‌ی شکل ۱-۱، دارای خطای $i > 0$ است[۱]. در صورت رخداد المان موردنظر در جایگاه صفر و مشاهده آن تنها یک‌بار در برنامه، کد موردنظر وجود آن را تشخیص نمی‌دهد که این خطا است.

ازآنجایی‌که پرهزینه‌ترین فعالیت‌های توسعه نرم‌افزار، هزینه کشف و اصلاح خطاها است به حداقل رساندن این خطاها و ساخت نرم‌افزاری عاری از خطا موضوعی پراهمیت در مهندسی نرم‌افزار است. برطرف کردن خطاهای نرم‌افزار با به‌کارگیری منابع آزمون امری پرهزینه است و اغلب از آن صرف‌نظر می‌شود. از سوی دیگر خطاهای ناشناخته و پیش‌بینی‌نشده موجود در اغلب نرم‌افزارها، هیچ خطایی در زمان کامپایل ندارند. لذا توسعه نرم‌افزار بدون خطا، حتی با به‌کارگیری بسیار دقیق روش‌های توسعه بسیار دشوار است. روند پیش‌بینی خطا، مسئولان و مدیران نرم‌افزار را قادر به تخصیص مقرون‌به‌صرفه منابع و کاهش هزینه‌های مربوط به آزمون می‌کند.

---

[۱] Fault

[۲] Failure

[۳] Error

```
/**
 * Find last index of element
 *
 * @param x array to search
 * @param y value to look for
 * @return last index of y in x; -1 if absent
 * @throws NullPointerException if x is null
 */
Public int findLast (int[] x, int y)
{
  for (int i=x.length-1; i>0; i--)
  {
    if (x[i] == y)
    {
        return i;
    }
  }
  return -1;
}
//test: x = [2, 3, 5]; y=2; Expected=0
//Book website: FindLast.java
//Book website: FindLastTest.java
```

شکل ۱-۱ نمونه خطای نرم‌افزار [۱]

کشف خطاهای نرم‌افزار در مراحل اولیه توسعه هزینه‌های کمتری را در پی دارد، بنابراین پیش‌بینی اولیه آن‌ها بسیار پراهمیت است. یافتن و از بین بردن تعداد خطاهای بیشتر سطوح بالاتری از کیفیت و قابلیت اتکای سیستم را فراهم می‌کند و تلاش برای نگهداری کیفیت را بهبود می‌دهد.

نرم‌افزار مطلوب نرم‌افزاری ایمن و باکیفیت است که به‌درستی به نیاز کاربر پاسخ بدهد، خطاهای پنهان در یک دوره زمانی مشخص و شرایط محیطی داده‌شده در آن وجود نداشته باشد و در زمان اجرای نرم‌افزار سیستم دچار شکست نشود. روند پیش‌بینی خطای نرم‌افزار[۱] یکی از روش‌های بهبود کیفیت است که از آموزش یک تکنیک یادگیری برای یک پروژه مشخص استفاده می‌کند[۲].

تشخیص اولیه و دقیق خطا و حذف زودهنگام آن در طول توسعه نرم‌افزار و قبل از عرضه به مشتری، توسعه یک سیستم بسیار قابل‌اعتماد و باکیفیت، کاهش تلاش و هزینه کلی آزمون، کاهش تأثیر خطاها و بهبود فرآیند توسعه اقتصادی بدون اجرا کردن کد نرم‌افزاری و پیش از انجام عمل آزمون از مزایای پیش‌بینی خطای نرم‌افزار است.

تکنیکی که درروند پیش‌بینی خطای نرم‌افزار مورداستفاده قرار می‌گیرد از رویکرد ساخت یک مدل پیش‌بینی خطا برای یک مجموعه‌داده در پروژه مشخص، پیروی می‌کند. سپس این مدل بر اساس

[۱] Software Fault Prediction

اطلاعاتی که در رابطه با نسخه‌های قبلی پروژه دارد آموزش می‌بیند و درنهایت در پیش‌بینی خطا برای پروژه‌های نامشخص مورداستفاده قرار می‌گیرد.

۱-۲ هدف تحقیق

پیش‌بینی خودکار خطاها و کشف دقیق آن‌ها یکی از چالش‌ها در حوزه کیفیت و قابلیت اتکای نرم‌افزار است. روش‌های بسیاری برای پیش‌بینی خطای نرم‌افزار به کار گرفته‌شده است که یکی از آن‌ها استفاده از شبکه‌های عصبی عمیق است. اخیراً اهمیت یادگیری عمیق، در مسائل پیش‌بینی خطا افزایش‌یافته است. این تکنیک به‌عنوان یک ابزار فرآیند پیش‌بینی را تسریع می‌کند. شبکه‌های عصبی عمیق به‌طور خودکار اطلاعات معنایی و ساختاری برنامه‌ها را از ویژگی‌های ورودی استخراج می‌کنند و در مقابل تغییرات ورودی مقاوم‌اند. علاوه بر تکنیک‌های مبتنی بر شبکه‌های عصبی عمیق، تکنیک‌های ترکیبی نیز برای پیش‌بینی خطای نرم‌افزار مؤثر واقع می‌شوند. هدف کلی این تحقیق بهبود دقت و عملکرد روش‌های پیش‌بینی خطا با استفاده از تکنیک یادگیری عمیق و روش ترکیبی است. به‌گونه‌ای که مدل پیش‌بینی با انجام خودکار فرآیند یادگیری، برای داده‌هایی که تاکنون ندیده است روند پیش‌بینی خطا را با دقت بالایی انجام دهد. همچنین استفاده از یک مجموعه‌داده بسیار خوب، به‌روز و منبع باز که در دسترس همه قرار دارد، ورودی‌های آن ساختار خوبی دارد و به فرآیند پیش‌بینی خطای نرم‌افزار در مقابل روش‌های قدیمی‌تر کمک می‌کند، دارای اهمیت ویژه‌ای برای انجام این کار است.

۱-۳ چالش‌ها و راه‌حل‌های آن

امروزه، با پیشرفت نرم‌افزارها، افزایش اندازه و پیچیدگی آن‌ها و فراهم آمدن شرایطی مطلوب برای رشد و ایجاد خطاهای نرم‌افزاری، رخداد این‌گونه خطاها غیرقابل‌اجتناب است. همچنین پیش‌بینی خودکار تعداد خطاها در واحدهای[۱] نرم‌افزار از اهمیت ویژه‌ای برخوردار است و در تخصیص منابع محدود مؤثر

[۱] Module

است[۳]. روش‌های مختلفی برای کشف خطاها و سپس از بین بردن آن‌ها ارائه‌شده است. اما بااین‌حال، دقت و عملکرد این رویکردها به پیشرفت قابل‌توجهی نیاز دارد.

تکنیک‌ها و الگوریتم‌های مختلف یادگیری ماشین[۱] که پیش‌ازاین جهت اعمال روند پیش‌بینی خطا به‌طور گسترده‌ای استفاده می‌شد، دقت‌ها و نتایج متفاوتی را ارائه داده است. از طرفی در مورد نرم‌افزارهای بزرگ، آزمایش تمام سناریوهای ممکن از طریق روش سنتی بسیار سخت و زمان‌بر است. یادگیری عمیق یک روش مناسب برای حل این مسئله است و عملکرد مدل‌های مبتنی بر یادگیری عمیق با افزایش داده، افزایش می‌یابد.

دقت و قابلیت شبکه یادگیری عمیق برای حل مسائل پیچیده به‌طور پیوسته افزایش می‌یابد. شبکه‌های عمیق عملکرد خوبی را در مسائل مشکل یادگیری و یادگیری ماشین ارائه می‌دهند. یادگیری عمیق[۲] به‌عنوان زیرشاخه‌ای از یادگیری ماشین، در حوزه‌ی پیش‌بینی خطای نرم‌افزار نتایج مطلوب‌تری را نسبت به تکنیک‌های سنتی ارائه می‌دهد. یادگیری عمیق همچنین عملکرد قابل‌توجهی را در زمینه‌های مختلف نظیر بینایی کامپیوتری، پردازش زبان طبیعی، تشخیص گفتار و سایر زمینه‌ها داشته است[۴].

اخیراً یادگیری عمیق برای افزایش دقت تکنیک‌های پیش‌بینی خطا مورداستفاده قرار می‌گیرد. مدل‌های یادگیری عمیق با استفاده از شبکه عصبی مصنوعی ساخته می‌شوند و معماری این شبکه‌ها متشکل از حداقل سه لایه است. افزایش تعداد لایه‌ها نشان‌دهنده پیچیدگی بیشتر مدل است. مدل‌هایی که توسط یادگیری عمیق ایجاد می‌شوند این توانایی را دارند که به‌صورت خودکار ویژگی‌های لازم را از داده استخراج کنند و در مقابل تغییرات ورودی مقاوم‌اند. در تکنیک‌های سنتی پیش‌بینی خطا که ویژگی‌ها به‌صورت دستی طراحی می‌شوند، استخراج اطلاعات ساختاری و معنایی برنامه‌ها اغلب امکان‌پذیر نیست و مدل‌های قدرتمند ایجاد نمی‌شوند. شبکه عصبی با کسب خودکار دانش از اطلاعات معنایی و ساختاری برنامه‌ها که آن‌ها را مستقیماً از ویژگی‌های مجموعه‌داده به دست می‌آورد، روند پیش‌بینی خطا را بهبود

---

[۱] Machine Learning

[۲] Deep Learning

می‌بخشد و مدل‌های دقیق پیش‌بینی را ایجاد می‌کند. تکنیک‌های یادگیری عمیق برای پیش‌بینی نیازی به استخراج ویژگی‌ها به‌صورت دستی ندارد[۵] و همواره شکاف بین معنای برنامه‌ها و ویژگی‌های پیش‌بینی خطا را پر می‌کند[۶]. این عمل پروسه‌ی پیش‌بینی خطا را بشدت تحت تأثیر قرار می‌دهد. جهت بهبود روند پیش‌بینی خطا وجود چنین رویکردی که پروسه پیش‌بینی را راحت‌تر، سریع‌تر، بهتر و با دقت بالاتر انجام دهد، الزامی است.

علاوه بر تکنیک‌های یادگیری عمیق، تکنیک‌های یادگیری ترکیبی نیز برای پیش‌بینی خطا بسیار مؤثر واقع می‌شوند و دقت مدل‌های پیش‌بینی خطا را نسبت به روش‌های منفرد بهبود می‌دهند. مدل‌های ایجادشده توسط این روش، با استخراج ویژگی‌های مختلف می‌توانند دقت مدل‌های پیش‌بینی را افزایش دهند و فرایند پیش‌بینی خطا را بهبود بخشند.

مسئله دیگری که در بحث پیش‌بینی خطا وجود دارد عدم وجود مجموعه‌داده‌های به‌روز و مرتبط با برنامه‌های نرم‌افزاری امروزی در بین مجموعه‌داده‌های فعلی پیش‌بینی خطای نرم‌افزار است. مجموعه‌داده باگ‌هانتر[۱] یک مجموعه‌داده جدید و بزرگ است که شامل برنامه‌های پراستفاده امروزی است. هریک از برنامه‌های این مجموعه‌داده دارای معیارهای نرم‌افزاری در سطح توابع است که می‌تواند پیش‌بینی خطای نرم‌افزار را بهبود دهد. چالشی که در این مجموعه‌داده وجود دارد عدم تعادل کلاس در نمونه‌های مجموعه است. جهت برقراری تعادل کلاس‌ها تکنیک رایج نمونه‌برداری تصادفی رو به بالا[۲] مورداستفاده قرار می‌گیرد که در نتایج پیش‌بینی خطا توسط شبکه‌های عصبی عمیق بسیار مؤثر است.

۱-۴ سؤال‌های تحقیق

آیا استفاده از یادگیری عمیق می‌تواند در پیش‌بینی خطای نرم‌افزار در سطح توابع مؤثر واقع شود؟

آیا می‌توان از روی معیارهای نرم‌افزاری، وجود خطا در یک تابع را به‌درستی پیش‌بینی کرد؟

---

[۱] BugHunter

[۲] Random Over Sampeling

آیا ایجاد مدل‌های پیش‌بینی با استفاده از یادگیری عمیق و یادگیری ترکیبی عملکرد مدل پیش‌بینی خطا را نسبت به رویکردهای پیشین بهبود می‌بخشد؟

۱-۵ فرضیه‌های تحقیق

با کمک معیارهای نرم‌افزاری موجود در مجموعه‌داده باگ‌هانتر و وجود داده‌های کافی می‌توان پیش‌بینی خطا را برای توابع انجام داد. مدل‌های یادگیری عمیق می‌توانند عملکرد فرایند پیش‌بینی خطای نرم‌افزار را نسبت به رویکردهای سنتی یادگیری ماشین بهبود بدهند. در کنار رویکردهای یادگیری عمیق، رویکردهای ترکیبی می‌توانند در روند پیش‌بینی خطای نرم‌افزار مؤثر واقع شوند. مدل‌های پیش‌بینی خطا با استفاده از روش‌های یادگیری عمیق و روش‌های ترکیبی، قابلیت تشخیص خطا برای داده‌های نامشخص را با عملکرد بالا خواهند داشت.

۱-۶ نوآوری‌های تحقیق

به‌طورکلی، لیست نوآوری‌ها در این پایان‌نامه به شرح زیر است:

- به‌کارگیری مجموعه‌داده خطای نوین باگ‌هانتر برای پیش‌بینی خطای نرم‌افزار که شامل برنامه‌های متنوع و محبوب است و امروزه توسط توسعه‌دهندگان و مهندسین نرم‌افزار بشدت مورداستفاده قرار می‌گیرد.
- متعادل‌سازی کلاس‌های مجموعه‌داده باگ‌هانتر توسط روش نمونه‌گیری تصادفی رو به بالا
- ارائه مدل خودکار پیش‌بینی خطا بر اساس شبکه عصبی پرسپترون چندلایه
- ارائه دو نوع مدل پیش‌بینی خطا جدید: یکی مبتنی بر شبکه عصبی پرسپترون چندلایه و یکی مبتنی بر مدل ترکیبی
- افزایش دقت مدل پیش‌بینی خطا نسبت به روش‌های قبلی