

علوم رایانش و فناوری اطلاعات

نشریه علمی انجمن کامپیوتر ایران

صاحب امتیاز: انجمن کامپیوتر ایران
مدیر مسئول: دکتر جعفر حبیبی
سردیبیر: دکتر احمد خونساری

شورای علمی

قاسم جابری پور، دانشیار دانشگاه شهید بهشتی
جعفر حبیبی، دانشیار دانشگاه صنعتی شریف
امیر حسین جهانگیر، دانشیار دانشگاه صنعتی شریف
شاهین حسابی، دانشیار دانشگاه صنعتی شریف
سید حمید حاجی سید جوادی، دانشیار دانشگاه شاهد
مسعود رهگذر، دانشیار دانشگاه تهران
مهدی صدیقی، دانشیار دانشگاه صنعتی امیرکبیر
和尚 فیلی، دانشیار دانشگاه تهران
عبدالرسول قاسمی، دانشیار دانشگاه خواجه نصربالدین طوسی
مصطفود عباسپور، دانشیار دانشگاه شهید بهشتی
محمد عبدالله ازگمی، دانشیار دانشگاه علم و صنعت ایران
مهدی کارگهی، دانشیار دانشگاه تهران
مازیار گودرزی، دانشیار دانشگاه صنعتی شریف
ناصر مزینی، دانشیار دانشگاه علم و صنعت ایران

همیدرضا ربیعی، استاد دانشگاه صنعتی شریف
حمید سربازی آزاد، استاد دانشگاه صنعتی شریف
کریم فائز، استاد دانشگاه صنعتی امیرکبیر
اکبر غفاری پور رهبر، استاد دانشگاه صنعتی سهند
احسان الله کبیر، استاد دانشگاه تربیت مدرس
کیوان ناوی، استاد دانشگاه شهید بهشتی
ناصر یزدانی، استاد دانشگاه تهران
محمد حسین یغمایی مقدم، استاد دانشگاه فردوسی مشهد
مرتضی آنانلویی، دانشیار دانشگاه علم و صنعت ایران
محسن ابراهیمی مقدم، دانشیار دانشگاه شهید بهشتی
حسین اسدی، دانشیار دانشگاه صنعتی شریف
احمد اکبری ازیرانی، دانشیار دانشگاه علم و صنعت ایران
رضا برنگی، دانشیار دانشگاه علم و صنعت ایران
حسین پدرام، دانشیار دانشگاه صنعتی امیرکبیر
نصرالله مقدم چرکری، دانشیار دانشگاه تربیت مدرس

همکاران دفتر نشریه

لیلا نورانی
مهردی دولتی

نشانی

تهران، خیابان آزادی، ضلع غربی دانشگاه صنعتی شریف، کوچه شهید ولی‌ا... صادقی، پلاک ۲۶، طبقه ۴، واحد ۱۶، دفتر انجمن کامپیوتر ایران، نشریه علوم رایانش و فناوری اطلاعات
تلفن: ۰۳۲۰۰۰-۸۷۲۲۴-۶۶۰۳۲۰۰
دورنگار: ۰۲۱۱۴۹۶

پست الکترونیکی: csitjour@gmail.com
نشانی سایت: <http://csi.org.ir/fa/publication/archive/name/csit>

مقالات درج شده در این نشریه صرفاً بیانگر نظرات مؤلفین آنها است و مسئولیت صحت و سقمه داده‌ها و نتایج بر عهده آنها است.

لیتوگرافی، چاپ و صحافی:

فهرست مقالات

۱	توصیه‌گر راه حل استثنائات در محیط یکپارچه ایجاد نرم افزار وحید امین‌تبار و عباس حیدرنوری
۱۶	آشکارسازی چهره انسان در تصاویر دیجیتال علیرضا زارعی و امین‌اله مه‌آبادی
۲۳	افزایش گذردهی در شبکه‌های موردنی به‌وسیله مدل صفت‌چابکی با در نظر داشتن احتمالات اختصاص اولیه محمد امین کشتکار، احمد خونساری و ابوالفضل دیانت
۳۲	شناسائی عملی کلیک‌های هرز در وب با استفاده از الگوریتم‌های دسته‌بندی مهریه فلاح و سجاد ظریف‌زاده
۴۰	طراحی و پیاده‌سازی جمع‌کننده دهدۀ افزونه‌ای با توان مصرفی پایین سعید گرگین و لیلا میرمقتدایی
۴۹	طراحی و مشخصه‌یابی بلوک منطقی قابل پیکربندی تمام گرافنی قابل استفاده در تراشه‌های FPGA سیدعلی سیف کاشانی و حسین کریمیان علی‌داش

توصیه‌گر راه حل استثنای در محیط یکپارچه ایجاد نرم‌افزار

وحید امین‌تبار عباس حیدرنوری

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

چکیده

در فرآیند ایجاد نرم‌افزار، استثناءها اجتناب‌ناپذیر هستند. ولیکن، استثناءها در بسیاری از موارد توسط برنامه‌نویسان دیگری نیز تجربه شده‌اند و به احتمال زیاد در فضای مجازی راه حل آن‌ها موجود است. با رخ دادن یک استثناء در زمان اجرا، معمولاً یک پیغام خطأ و موارد بالای پشته فراخوانی^۱، نمایش داده می‌شوند. با گسترش استفاده از سایتهاهی پرسش و پاسخ برنامه‌نویسی مانند Stack Overflow و یا سایتهاهی حاوی کدهای متن باز مانند Source Forge، برنامه‌نویسان اغلب برای پیدا کردن راه حل یک استثناء، با استفاده از مرورگر و موتورهای جستجو و به دنبال مشکلات و راه حل‌های مشابه می‌گردند. در اینجا شکاف بین محیط ایجاد نرم‌افزار و مرورگر وجود دارد. همچنین جستجوی دستی، تمام اطلاعات کد برنامه‌نویس را دارا نمی‌باشد و کاری زمان بر است. در این مقاله قصد آن داریم تا Exception Tracer را معرفی نماییم. در این روش با استفاده از پشته فراخوانی مربوط به استثناء و کد برنامه‌نویس ابتدا در پروژه‌های Source Forge فایل‌های مرتبط بازیابی می‌شوند و در ادامه با ساخت گراف GROUM از کد برنامه‌نویس و فایل‌های مرتبط، قطعه کدها رتبه‌بندی می‌شوند و بعد از تغییر نام متغیرهای آن‌ها به زمینه برنامه‌نویس، قطعه کدهایی که می‌توانند حاوی راه حل استثناء باشند، به او پیشنهاد داده می‌شوند. GROUM گرافی است که در آن توالی فراخوانی متدها و واسطگی دادهای بین آن‌ها را شامل می‌شود و نتیجه بهتری نسبت به جستجوی متنی و یا درخت تحوی می‌دهد. این روش، علاوه بر قطعه کد، مباحثه‌های مرتبط را نیز از Stack Overflow استخراج می‌کند و به برنامه‌نویس نمایش می‌دهد. روش پیشنهادی این مقاله، به صورت یک افزونه Eclipse پیاده‌سازی شده است. دقت این روش با استفاده از تعدادی از نمونه کدهای مربوط به کتابخانه‌های Apache جوا اندازه‌گیری شده است. در ارزیابی‌های انجام شده، در ۷۵ درصد مواقع، راه حل استثناء رخداده در ۵ نتیجه ابتدایی وجود دارد. علاوه بر محاسبه دقت، میزان صرف‌جویی زمانی نسبت به استفاده ساده از مرورگر نیز از طریق مطالعه کاربران سنجیده شده است. طبق این ارزیابی، میزان صرف‌جویی زمانی، در حدود ۳۸ درصد است.

کلمات کلیدی: محیط یکپارچه ایجاد نرم‌افزار، سیستم‌های توصیه‌گر، قطعه کد، Stack Overflow، افزونه Eclipse.

۱- مقدمه

را بینديم. عدم رعایت توالی ذکر شده، منجر به استثناء خواهد شد. به علت سرعت چشم‌گیر ایجاد و به روزرسانی چارچوب‌ها و کتابخانه‌ها، مستندات کافی و به روز برای آن‌ها وجود ندارد. به دلیل عدم وجود مستندات مناسب، ممکن است برنامه‌نویس به اشتباه از کتابخانه‌ها استفاده کند و اجرای کد به استثناء منجر شود. وقتی که یک استثناء رخ می‌دهد، برنامه‌نویسان اغلب از محیط یکپارچه ایجاد نرم‌افزار به مرورگرهای وب می‌روند و با جستجوی پیغام استثناء در موتورهای جستجوی عمومی به دنبال راه حل مشکل خود می‌گردند. ولیکن، برنامه‌نویس معمولاً در ایجاد پرس‌وجو فقط از پیغام خطأ استفاده می‌کند و اطلاعات زمینه کد را دخیل نمی‌کند. علاوه بر آن، برنامه‌نویس زمان نسبتاً زیادی را برای یافتن پاسخ به وسیله مرورگر و دیدن صفحات وب سپری می‌کند. مطالعات نشان می‌دهد که ایجاد‌کنندگان نرم‌افزار در طول فرآیند ایجاد و نگهداری نرم‌افزار، ۱۹٪ زمان خود را

مهندسي نرم‌افزار به سرعت در حال پیشرفت است. روزانه چارچوب‌ها^۲ و کتابخانه‌های برنامه‌نویسی جدیدی ایجاد و یا به روزرسانی می‌شوند، و استفاده از آنها به دلیل افزایش سرعت و کیفیت برنامه‌نویسی به سرعت در حال گسترش است. ولیکن، فهم چگونگی استفاده از رابط کاربردی برنامه‌نویسی^۳ این کتابخانه‌ها و چارچوب‌های برنامه‌نویسی سخت و زمان بر است. برنامه‌نویس برای استفاده از آن‌ها باید یک دنباله مشخصی از فراخوانی متدها را دنبال کند؛ عدم رعایت این قوانین ممکن است منجر به استثناء شود. به عنوان نمونه، برای نوشتن در فایل باید ابتدا فایل را باز کنیم؛ سپس داده موردنظر را در فایل بنویسیم و در انتهای نیز فایل

استفاده برنامه‌نویسان، (یعنی، استفاده از مروگر) مقایسه شود، روش پیشنهادی را به دو روش متفاوت ارزیابی نمودیم: (۱) ارزیابی کارایی و دقت؛ (۲) مطالعه کاربران در روش اول ارزیابی که به روی ۱۴ کتابخانه از پروژه‌های Apache و کتابخانه‌های استاندارد Java اجرا گرفت، خروجی روش پیشنهادی از دقت ۷۵ درصدی برخوردار بود، بدین معنی که برنامه‌نویس در ۷۵ درصد موارد در نتیجه اول جواب خود را می‌یابد. با توجه به نتایج ارزیابی دوم، برنامه‌نویسان با استفاده از Exception Tracer نسبت به حالتی که از مروگر استفاده نمایند، ۳۸ درصد در زمان خود صرفه‌جویی می‌نمایند.

با توجه به بحث فوق، نوآوری‌های ما در این مقاله عبارتند از:

- ایجاد امکان یافتن راحل استثناء در محیط یکپارچه ایجاد نرم‌افزار و بدون استفاده از مروگر.
- پیاده‌سازی روش پیشنهادی به صورت یک افزونه برای نرم‌افزار Eclipse.
- ایجاد خودکار پرس‌وحو از زمینه کد برنامه‌نویس برای موتور جست‌وجوی BOA.
- رتبه‌بندی و تبدیل قطعه‌کدهای استخراجی از Source Forge به زمینه برنامه‌نویس.

خواننده‌گان علاقمند، برای مشاهده محیط این ابزار می‌توانند ویدئو موجود در آدرس زیر را مشاهده نمایند:

<https://goo.gl/q7Fr56>

در ادامه این مقاله، در بخش دوم کارهای مرتبط را بررسی می‌کنیم. در بخش سوم، مسئله‌ای که قصد حل کردن آن را داریم را به صورت دقیق بیان می‌کنیم. سپس در بخش چهارم، به بیان راحل پیشنهادی برای حل این مسئله می‌پردازیم. در بخش پنجم، به ارزیابی راحل پیشنهادی پرداخته خواهد شد. در نهایت، در بخش ششم، به بیان کارهای آتی، و در بخش هفتم، به نتیجه‌گیری می‌پردازیم.

۲- کارهای مرتبط

در این بخش کارهای مرتبط در دو دسته سیستم‌های توصیه‌گر و استخراج مشخصات تقسیم‌بندی شده‌اند. این دسته‌بندی‌ها هر کدام شامل حوزه‌های بسیار گسترده‌ای می‌شوند. در این مقاله فقط مقالاتی از این دسته‌ها که شباهتی با روش پیشنهادی داشته‌اند مورد بررسی قرار گرفته‌اند. روش پیشنهادی در این مقاله ترکیبی از روش‌های سیستم‌های توصیه‌گر و روش‌های استخراج مشخصات است.

۲-۱- سیستم‌های توصیه‌گر

موارد بررسی شده در این دسته به طور معمول به صورت افزونه Eclipse پیاده‌سازی شده‌اند و به شکل‌های مختلف به برنامه‌نویس پیشنهادهایی را برای انجام بهتر برنامه‌نویسی ارائه می‌دهند. به عنوان مثال، سیستم‌های توصیه‌گر نمونه کد، با توجه به اطلاعات استخراج شده از محیط برنامه‌نویسی و جست‌وجو در منابع کد به منظور بهبود فرآیند برنامه‌نویسی به برنامه‌نویس نمونه کدهایی را پیشنهاد می‌دهند. روش‌های [۱۱-۱۸] از این قبیل روش‌ها هستند. از دیگر سیستم‌های توصیه‌گر می‌توان [۱۹-۲۲] را نام برد که در این مقالات پیشنهاد روش‌های بازار آرایی^۵ مدنظر است. به عنوان یک نمونه دیگر می‌توان [۱۴] را نام برد که نمونه کد خاص واحد آزمون پیشنهاد می‌دهد. گروهی دیگر از سیستم‌های توصیه‌گر هستند که به منظور اصلاح خطای موجود در کد برنامه‌نویس طراحی شده‌اند. روش پیشنهادی این مقاله نیز همین هدف را دارد. با توجه به ارتباط زیاد این دسته از کارها به این مقاله، در ادامه ۶ روش از این حوزه، به تفصیل توضیح داده شده‌اند.

به جست‌وجوی اطلاعات با استفاده از موتورهای جست‌وجوی رایج می‌گذرانند [۱]. همچنین تغییر محیط به مروگر از محیط یکپارچه ایجاد نرم‌افزار به عدم تمکز برنامه‌نویس منجر می‌شود. تاکنون روش‌های مختلفی برای حل این مسئله مطرح شده است [۱-۶]. روش‌های [۱-۲] فقط از پشته فراخوانی در جست‌وجو استفاده می‌کنند. هیچ کدام از این روش‌ها یک نوع خروجی می‌دهند. این خروجی‌ها یکی از موارد قطعه کد، مباحثه و یا صفحات وب است. روش‌های [۲-۴، ۶] فقط مباحثه‌های مرتبط را به عنوان خروجی به برنامه‌نویس ارائه می‌دهند؛ روش [۳] مباحثه و نمونه کد خروجی می‌دهد ولی نمونه کد آن در حد یک خط است. روش [۵] نیز صفحات وب را ارائه می‌دهد. هر کدام از این خروجی‌ها (کد و یا مباحثه) بخشی از استثناءها را پوشش می‌دهند. همچنین، در تمام این روش‌ها نمونه کدهای خروجی ابزارهای مشابه به زمینه کد برنامه‌نویس منتقل نشده‌اند. برنامه‌نویس به منظور استفاده از این نمونه کدها، مجبور است اسامی متغیرها را به صورت دستی به اسامی موجود در کد خود تغییر دهد. در این مقاله، منظور از انتقال به زمینه برنامه‌نویس، تغییر نام متغیرهاست به گونه‌ای که اسامی متغیرها با کدی که برنامه‌نویس در حال کارکردن بر روی آن است، سازگار باشد.

به منظور حل مشکلات فوق، در این مقاله، ابزاری با عنوان Exception Tracer ارائه شده است. این ابزار، به صورت خودکار، به حل استثناء در زبان جاوا به بخش مباحثه‌های مرتبط به آن استثناء را از سایت پرسش و پاسخ Stack Overflow و قطعه کدهایی که ممکن است راحل استثناء باشد را از سایت Source Forge به برنامه‌نویس پیشنهاد دهد. برای یافتن مباحثه‌های مرتبط از سایت پرسش و پاسخ Stack Overflow، از اطلاعات موجود در پشته فراخوانی استفاده می‌شود. برای یافتن نمونه کدهای مرتبط، ابتدا با استفاده از پشته فراخوانی، رابط کاربردی برنامه‌نویسی مورد استفاده کاربر را شناسایی می‌کند؛ فایل‌های شامل این رابط کاربردی برنامه‌نویسی به وسیله زبان پرس‌وحوی BOA [۳] از پروژه‌های Source Forge بازیابی می‌شوند. یک موتور جست‌وجو است که جست‌وجو را به صورت موازی به روی درخت نحو پروژه‌های Source Forge انجام می‌دهد. سپس یک دید سطح بالا (گراف) از این کدها و کد برنامه‌نویس ایجاد می‌شود. بعد از آن، با استفاده از گراف GROOM [۷]، قطعه کدها به زمینه کاربر انتقال داده می‌شوند. در نهایت، از این گراف‌ها کد تولید می‌شود. با توجه به میزان شباهت گراف قطعه کدهای تولید شده به گراف برنامه برنامه‌نویس، آن قطعه کدها رتبه‌بندی می‌شوند و به برنامه‌نویس نمایش داده می‌شوند. Exception Tracer با هدف ارائه پیشنهاد برای آسان‌سازی اصلاح کدهایی که قوانین استفاده از رابط کاربردی برنامه‌نویسی در آن‌ها رعایت نشده‌اند، ایجاد شده است. روش‌های مشابه، در جست‌وجوهایشان در نهایت با کد به صورت متنی برخورد می‌کنند. روش ارائه شده در این مقاله، برای جلوگیری از این امر، یک دید سطح بالا از کد نیز ایجاد می‌شود. این دید سطح بالا باید شامل قوانین استفاده از رابط کاربردی برنامه‌نویسی (معمولاً به شکل دنباله‌ای از فراخوانی متدها) باشد. برای استخراج دنباله فراخوانی متدها از کد، از گرافی که از درخت انتزاعی نحو^۶ برنامه ساخته شده، استفاده می‌شود و الگوهای استفاده را شامل می‌شود. این گراف با توجه به وایستگی‌های داده‌ای موجود در کد ایجاد می‌شود. با روش ارائه شده، اولاً، کاربر برای یافتن راحل نیازی به خروج از محیط یکپارچه ایجاد نرم‌افزار ندارد. ثانیاً، اطلاعات زمینه کد کاربر در پرس‌وحو درگیر است. ثالثاً، کاربر زمان کمتری را به جست‌وجو می‌پردازد. رابعاً، از قطعه کد پیشنهادی دستورات نامرتبط حذف می‌شوند و خروجی نهایی در زمینه یک در برنامه‌نویس است؛ به همین دلیل فهم و استفاده از آن آسان‌تر صورت می‌پذیرد.

به دلیل آن که روش‌های مشابه این مقاله با معیارهای گوناگونی ارزیابی شده‌اند و با توجه به آنکه باید Exception Tracer با آن‌ها و نیز روش معمول مورد

Oscilloscope - ۱-۱-۲

ابزار Oscilloscope [۲] برای شناسایی خطاهای مشابه از پشتی فرآخوانی استفاده کرده است. برای یک خطای جدید، با استفاده از پشتی فرآخوانی در پایگاه داده خود جستجو می‌کند و خطاهای مشابه را پیدا می‌کند. سپس به برنامه‌نویس گزارش‌های آن خطاهای را نمایش می‌دهد. این روش به صورت افزونه Eclipse پیاده‌سازی شده است. وقتی یک JUnit منجر به خطا می‌شود، نتیجه آزمون به سرو برگذاری می‌شود. با استفاده از اطلاعات برگذاری شده، در پایگاه داده جستجو انجام می‌گیرد و نتیجه (لیستی از گزارشات خطای) به برنامه‌نویس نمایش داده می‌شود. از آنجایی که پشتی فرآخوانی ممکن است بزرگ باشد و جستجو زمان بر شود، برنامه‌نویس می‌تواند قسمتی از پشتی فرآخوانی را برای جستجو انتخاب کند. در مقایسه با Exception Tracer این ابزار قطعه کد پیشنهاد نمی‌دهد و زمینه کدی که منجر به استثناء شده است را در جستجو دخیل نمی‌کند.

Help Me Out - ۲-۱-۲

ابزار Help Me Out [۳] از سه بخش اصلی تشکیل شده است: (۱) افزونه Eclipse پایگاه داده شامل خطاهای راه حل‌هایشان (۲) واسط کاربری برای اضافه کردن توضیح به راه حل‌ها. در این روش تغییرات کد برنامه‌نویس مورد بررسی قرار می‌گیرد. هنگامی که برنامه از حالت خطدار به حالت بدون خطا می‌رود، تغییرات کد به عنوان راه حل برای خطای رخداده، در نظر گرفته می‌شود و در پایگاه داده ذخیره می‌شود. هنگامی که خطای در برنامه رخ می‌دهد، در پایگاه داده براساس پیغام خطای پشتی فرآخوانی، خطی از کد که خطای در آنجا رخداده و رای کاربران به راه حل‌ها، در پایگاه داده جستجو انجام می‌گیرد. یک مرحله مهم، تشخیص گذار از حالت خطدار به حالت بدون خطا در کد است. دو مورد خطا داریم: (۱) خطای زمان کامپایل (۲) خطای زمان اجرا. در مورد اول اگر در کامپایل بعدی خطای ندهد گذار به راحتی تشخیص داده می‌شود. اما در مورد دوم عدم رخدادن خطای زمان اجرا ممکن است به دلیل متفاوت بودن ورودی‌های برنامه باشد. برای کمتر شدن رخداد این اشتباه، تعداد دفعات اجرای خطی که در آن قبل از خطا را نگه داری می‌کنیم. در صورتی که در اجرای بعدی در آن خط خطا رخداد و به علاوه تعداد دفعات اجرا شدن آن خط با اجرای قبلی یکسان بود، گذار تشخیص داده می‌شود. فقط در مورد اول (خطای زمان کامپایل) در جستجو، کد برنامه‌نویس (فقط تک خطی که خطای در آن رخداد با تعمیم) در نظر گرفته می‌شود. در مورد دوم فقط پشتی فرآخوانی برای جستجو استفاده می‌شود. در نهایت راه حل به برنامه‌نویس نمایش داده می‌شود. برنامه‌نویس می‌تواند به راه حل رای دهد و از این اطلاعات برای جستجوی بهتر برهه برده می‌شود. این ابزار در ابتدای استفاده از آن و وقتی که پایگاه داده هنوز پر نشده است، پیشنهادی نمی‌دهد، در حالی که این گونه نیست.

Surf Clips - ۳-۱-۲

روش Surf Clips [۵] نتایج جستجو در چهار وبسایت Google، Bing، Yahoo و Stack Overflow رتبه‌بندی و به برنامه‌نویس نمایش می‌دهد. با رخدان استثناء، پرس‌وجو براساس کد و پشتی فرآخوانی ساخته می‌شود. پرس‌وجو از کلمات زیر ساخته می‌شود: (۱) پنج نام کلاس یا متاد با درجه جذابیت بالاتر (۲) پنج پر تکرار ترین نام متاد فرآخوانی شده یا کلاس (موجود در import) از کد برنامه‌نویس. سپس نتایج موتورهای جستجو جمع‌آوری و براساس معیارهای زیر

Prompter - ۶-۱-۲

ابزار Prompter [۶] تغییرات کد برنامه‌نویس را رصد می‌کند. زمینه‌ی کدهای جدید به سازنده پرس‌وجو فرستاده می‌شود. در صورتی که پرس‌وجو جدید ساخته شود، به همراه زمینه‌ی کد به موتور جستجو ارسال می‌شود.

۴-۱-۲- ابزار ارائه شده توسط کوردیرو

ابزار ارائه شده توسط کوردیرو [۱] دو بخش اصلی دارد: (۱) خدمتگزار؛ (۲) افزونه Eclipse. در خدمتگزار یک پایگاه داده از داده‌های Stack Overflow درست شده است. با رخدادن استثناء اطلاعات پشتی فرآخوانی به خدمتگزار ارسال می‌شود و خدمتگزار مرتبط‌ترین مباحثت به آن را بر می‌گرداند. هر مباحثتی علاوه بر متن، ممکن است دارای کد و یا پشتی فرآخوانی باشد؛ قطعه کدهای کوچک در نظر گرفته نمی‌شوند. قطعه کدهای بزرگ تجزیه می‌شوند و ارجاع‌های آن‌ها به دست می‌آید. ارجاع‌های پشتی فرآخوانی نیز با عبارت منظم از کد تمیز داده می‌شود. منظور از ارجاع، رخداد یک نوع است. این اطلاعات به وسیله موتور جستجوی Apache Lucerne نمایه‌سازی می‌شوند.

با ارسال اطلاعات پشتی فرآخوانی، خدمتگزار یک پرس‌وجو شامل نام تمام استثناء‌های موجود در پشتی فرآخوانی و متن پنج ارجاع بالای پشتی فرآخوانی را می‌سازد. از نتایج به دست آمده دویست موردی که در عنوان آن‌ها نام تمام استثناء‌ها آمده باشد، برگردانده می‌شود و به برنامه‌نویس نمایش داده می‌شود. در مقایسه با Exception Tracer این ابزار قطعه کد ارائه نمی‌دهد و همچنین زمینه کد را در جستجو دخیل نمی‌کند.

Seahawk - ۵-۱-۲

ابزار Seahawk [۱۵] به صورت خودکار پرس‌وجو را می‌سازد و با جستجو در داده‌های Stack Overflow نتایج به برنامه‌نویس نمایش داده می‌شود. برنامه‌نویس علاوه بر مشاهده مباحثه می‌تواند در کد به آن ارجاع دهد تا بقیه افراد گروه نیز بتوانند آن را مطالعه کنند.

داده‌ها در یک پایگاه داده رابطه‌ای ذخیره شده‌اند و جستجو به وسیله انجام می‌شود. برای ساخت پرس‌وجو کدهای نزدیک اشاره‌گر Apache Solar در نظر گرفته می‌شوند. اعمال بازیابی اطلاعات روی این کد انجام می‌شود و ۱۰ کلمه پر تکرار آن استخراج می‌شود. همچنین از بخش import کلمات استخراج می‌شوند. اجتماع این دو مجموعه پرس‌وجو را می‌سازد. نتیجه این پرس‌وجو به برنامه‌نویس نمایش داده می‌شود. این ابزار در جستجو پشتی فرآخوانی را دخیل نمی‌کند و برای خطاهای مبتنی بر استثناء مناسب نیست.

جدول ۱- مقایسه سیستم‌های توصیه‌گر موجود برای اصلاح خطاهای برنامه‌نویسی

نام ابزار	برنامه‌نویس	انتقال به زمینه	داده‌های ورودی	استفاده از کد برنامه‌نویس در جستجو	پیش‌پردازش به روی کد داده‌های پایگاه داده	فرآخوانی	استفاده از پشتۀ تجزیه‌ای کد
Oscilloscope	-	-	منابع گزارش خطای پایگاه‌داده شخصی	-	-	-	☒
Help Me Out	☒	☒	صفحات وب	☒	☒	☒	☒
Surfclipse	-	-	Stack Overflow	-	☒	-	☒
ابزار کوردیرو	-	-	Stack Overflow	-	☒	☒	☒
Seahawk	-	-	Stack Overflow	-	☒	-	-
Prompter	-	-	Stack Overflow	-	☒	-	-
Exception Tracer	☒	☒	کدهای مخازن نرم‌افزاری	☒	-	☒	☒

هیچ کدام از این ابزارها پیشنهاد اذنشان را به زمینه کاربر به طور کامل منتقل نمی‌کنند.

۲. داده‌های ورودی: هر یک از ابزارهای معرفی شده با جستجو به روی داده‌هایی به برنامه‌نویس پیشنهاد ارائه می‌دهد از این داده‌ها می‌توان صفحات وب، اطلاعات Stack Overflow را نام برد. ابزار ارائه شده توسط کوردیرو، Stack و Promter تنهای از Stack Overflow استفاده می‌کنند. روش عمومی‌تری ارائه کرده است و جستجو را به روی Oscilloscope می‌شود. هر چه آن‌روپی یک کلمه بیشتر باشد، ارزش کمتری دارد. بعد از ساخته شدن پرس‌وجو و بازیابی مباحثه‌ها از متورهای جستجو، باید آن‌ها را رتبه‌بندی کنیم. رتبه‌بندی براساس ۸ ویژگی زیر انجام می‌شود:

۱. شباهت متنی بین کد برنامه‌نویس و متن (غیر کد) مباحثه.
۲. درصد خطاهایی از کد برنامه‌نویس که با قسمتی از مباحثه شباهت دارد.
۳. نوع‌هایی که در کد برنامه‌نویس و مباحثه وجود دارند.
۴. متدهای فراخوانی شده در کد برنامه‌نویس که در مباحثه آمده است.
۵. امتیازی که کاربران Stack Overflow به مباحثه داده‌اند.
۶. مباحثه‌ای که جواب تایید شده داشته باشد، ارزش بیشتری دارد.
۷. تعداد پاسخ‌های مباحثه، در امتیازدهی تاثیرگذار است.
۸. تعداد برچسب‌های مباحثه که با کلمات به دست آمده از بخش Import کد برنامه‌نویس یکسان‌اند.

۳. استفاده از کد برنامه‌نویس در جستجو: در جستجو می‌توان از پشتۀ فراخوانی و کد برنامه‌نویس بهره جست. این معیار استفاده از کد برنامه‌نویس در جستجو را بررسی می‌نماید. ابزار ارائه شده توسط کوردیرو و Oscilloscope برخلاف بقیه ابزارها در جستجو، کد برنامه‌نویس را دخیل نمی‌کنند. داده‌هایی که جستجو در آن‌ها صورت می‌گیرد، ممکن است شامل کد باشند؛ ابزارهای Oscilloscope و Surfclipse برای این بخش از اطلاعات تدبیر خاصی در نظر نگرفته‌اند و مانند بقیه متن‌ها با آن‌ها برخورد می‌کنند. اما بقیه ابزارها یک پیش‌پردازشی به روی این بخش از اطلاعات انجام می‌دهند تا جستجو با کیفیت بهتری انجام شود. در کل همه ابزارها، جستجوهایشان در انتهای به جستجو متنی انجام می‌گیرد.

Exception Tracer برش خود را به صورت گرافی انجام می‌دهد و کد برنامه‌نویس در جستجو تأثیر مستقیم دارد. ۴. پیش‌پردازش به روی کدهای پایگاه داده: برای جستجوی دقیق‌تر و سریع‌تر برخی ابزارها به روی داده‌های ورودی خود پیش‌پردازش انجام می‌دهند. این معیار این مورد را مشخص می‌نماید. برخی از ابزارها در جستجو با کد مانند متن برخورد می‌کنند (Oscilloscope، Surf Clips و Seahawk) و برخی دیگر برای افزایش دقت در جستجو یک پیش‌پردازشی (Seahawk) به روی کد انجام می‌دهند و در نهایت همان جستجوی متنی را انجام می‌دهند، ابزار کوردیرو و Help Me Out (Seahawk).

مباحثه‌ها براساس معیارهای بالا رتبه‌بندی می‌شوند و در صورتی که مباحثه‌ای وجود داشته باشد که از یک حدی امتیازش بیشتر باشد، به برنامه‌نویس نمایش داده می‌شود. این ابزار در جستجو پشتۀ فراخوانی را دخیل نمی‌کند و برای خطاهای مبتنی بر استثناء مناسب نیست.

۷-۱-۲- مقایسه سیستم‌های توصیه‌گر موجود برای اصلاح خطاهای برنامه‌نویسی

تاکنون روش‌های مرتبط به تفصیل بیان شدند. در این بخش این روش‌ها را با چند معیار مختلف با یکدیگر مقایسه می‌کنیم (جدول ۱). این معیارها با توجه به مراحل این ابزارها انتخاب شده‌اند. تمام این ابزارها شامل داده‌های ورودی، مرحله جستجو و نمایش نتیجه هستند. ولیکن، هر کدام به نحوی این مراحل را دارا هستند:

۱. انتقال به زمینه کد برنامه‌نویس: این معیار بیان می‌کند که آیا ابزار به روی خروجی خود پس پردازشی برای انتقال به زمینه کد برنامه‌نویس انجام می‌دهد یا خیر. از بین ابزارهای معرفی شده تنها Help Me Out مانند Exception Tracer به برنامه‌نویس راه حل را به زمینه کاربر منتقل می‌کند. این انتقال سیار محدود است و در حد یک خط از کد، انجام می‌گیرد و لیکن

کتابخانه‌ای از Eclipse استخراج می‌شود. سپس با پیمایش آن، گراف ایجاد می‌شود. این گراف از نوع گراف جهت دار بدون دور (DAG) است و می‌توان به خاطر همین خاصیت، روند استفاده از یک API را به راحتی از آن استخراج نمود. راس‌ها در این گراف دو نوع هستند: (۱) رأس کنشی (فراخوانی متدهای دسترسی به Field و فراخوانی Constructor) (۲) رأس کنترلی (بیان کننده ساختار کنترلی while). یال‌ها در این گراف دو نوع هستند؛ برخی یال‌ها توالی خطهای کد مربوط به هر رأس را مشخص می‌کند؛ برخی دیگر وابستگی داده‌ای را نشان می‌دهند. رأسی که یال وابستگی داده‌ای به آن وارد شده است، به رأس ابتدای یال وابستگی داده‌ای دارد و از داده‌ی مهیا شده توسط رأس ابتدای یال استفاده می‌کند.

۲-۲-۲ Jungloid

در این روش یک فرض وجود دارد. "برنامه‌نویس نوع شی نهایی موردنیاز خود را می‌داند اما نمی‌داند این شی را چگونه به دست آورد [۲۰]." در واقع ورودی این روش کد به همراه نوع خروجی است. در نهایت قطعه کد مربوط به این سؤال به برنامه‌نویس نمایش داده می‌شود. ابتدا از کد، یک گراف ساخته می‌شود. رأس‌های این گراف نوع‌های کلاس موجود در API هستند. یال‌ها ممکن است یکی از موارد زیر باشند:

۱. دسترسی به Field

۲. فراخوانی Static متدهای Static و یا Constructor

۳. فراخوانی متدهای روی یک شی.

۴. ارجاع از شیء فرزند به پدر (Up cast).

۵. ارجاع از شیء پدر به فرزند (Downcast).

بعد از رسم گراف کد مربوط به API، مسیرهای منتهی به نوع موردنظر بررسی می‌شوند. مسیرهایی که دور ندارند، جواب ما هستند. در نهایت این گراف به کد تبدیل می‌شود و به برنامه‌نویس نمایش داده می‌شود. روش‌های دیگری نیز برای هدف Jungloid که در بالا ذکر شد، ارائه شده است. Snippet X [۲۱]، API Synth PARSE Web [۲۲] و API Synth [۲۳] سه روش دیگر از این دسته هستند. از آنچه این دسته برای پیاده‌سازی انتخاب نشده است، وارد جزئیات تفاوت ابزارهای این دسته نشده‌ایم.

۲-۳-۲ روش انتخابی

این روش‌ها از کد، گراف بدون دور DAG ایجاد می‌کنند و پردازش‌هایی روی آن گراف‌ها انجام می‌دهند. نوع و نحوه کشیدن گراف در این روش‌ها برای ما اهمیت دارد. در روش‌های یاد شده، به‌جز GROUM فرض بر این است که می‌خواهیم از یک قطعه کد، روش ساختن شی از نوع داده‌ای موردنظر را پیدا کنیم. در این روش‌ها، ورودی یک قطعه کد به همراه یک نوع داده‌ای است و خروجی آن یک دنباله از فراخوانی متدهایی است که به شی از نوع موردنظر ختم می‌شود. اما GROUM این فرض را ندارد و یک گراف با جزئیات بیشتر از کد ترسیم می‌کند. GROUM برای جلوگیری از بزرگ شدن، گراف‌هایش را درون متدهای ایجاد می‌کند. به این معنی که برای هر متدهایی یک گراف جدا کشیده می‌شود و وابستگی‌های فرامت در نظر گرفته نمی‌شوند. همچنین در گراف GROUM ساختار شرطی نیز وجود دارد، اما در مابقی روش‌ها خیر. برای پاسخ‌گویی در زمان مناسب، نیاز دارد تا نوع شی ورودی نیز تعیین شود. به عبارت دیگر، ورودی روش، کد، نوع شی ورودی و نوع شی نهایی خواهد بود و بین این دو رأس در گراف به دنبال کوتاه‌ترین مسیر می‌گردد. از طرفی در Jungloid کوتاه‌ترین مسیر به عنوان جواب

جستجویش را به روی درخت تجزیه کد انجام می‌دهد و ساختار کد در جستجو دخیل می‌شود.

۵. استفاده از پشتۀ فراخوانی در جستجو: این معیار استفاده از پشتۀ فراخوانی در جستجو را مورد بررسی قرار می‌دهد. تمامی ابزارهای به غیر از Seahawk و Prompter در جستجوهای این اطلاعات موجود در پشتۀ Seahawk پایگاه داده‌هایی از محتویات دیگر و بگاه‌ها استفاده کردند. ابزار ارائه شده توسط کاربران Help Me Out امکان ورود اطلاعات از دیگر وبگاه‌ها را ندارد، اما خوبه‌خود اطلاعاتش توسط کاربرانش به روزرسانی می‌شود. ابزارهای Surfclipse و Prompter جستجو را به صورت برخط انجام می‌دهند و از این نظر اطلاعاتشان بدون نیاز به انجام کاربرانش به روزرسانی می‌شوند. علاوه بر این، پایگاه داده آن نیز ارتقا می‌یابد.

۶. جستجوی تجزیه‌ای کد: عدم جستجوی متنی هدف این معیار است. همان‌طور که در معیار قبلی توضیح داده شد، تمامی این ابزارها درنهایت کد را به صورت متن در نظر می‌گیرند و جستجوی متنی انجام می‌دهند. اما با مقایسه گراف‌های تولید شده از کد ساختار کد را به طور کامل در جستجو دخیل می‌کند.

۲-۲-۲ استخراج مشخصات

حوزه این روش‌ها بسیار گسترده می‌باشد و کاربردهای گوناگونی دارند. ولیکن هدف کلی این روش‌ها شناسایی پروتوكلهایی است که برنامه‌های کاربران باستانی رعایت کنند و قوتی که از یک کتابخانه و یا چارچوب نرم‌افزاری استفاده می‌کنند. این روش‌ها به صورت عمومی به دو دسته پوپا و ایست تقسیم می‌شوند. روش‌های پوپا به اجرای برنامه نیاز دارند و داده‌های خود را از داده‌های زمان اجرا استخراج می‌کنند. به عنوان نمونه می‌توان به روش‌های [۱۶-۱۸] اشاره نمود. روش [۱۹] ابتدا به صورت پوپا قوانین را استخراج می‌کند، ولیکن بررسی عدم‌راعایت قوانین در کد برنامه‌نویس به صورت ایستا انجام می‌گیرد. روش‌های ایستا به اجرای برنامه‌ی برنامه‌نویس نیازی ندارند و با تحلیل کد او قوانین را استخراج می‌کنند. استفاده غلط از یک رابط کاربردی برنامه‌نویسی ایجاد شده است. ضعف ابزارهای مشابه در جستجوی متنی به روی کد است. به همین دلیل روش ارائه شده در این مقاله به یک دید سطح بالا از کد نیاز دارد تا جستجو را با دقت بهتری انجام دهد. مهم‌ترین خصوصیت این دید سطح بالا این است که دنباله فراخوانی را نمایش دهد؛ زیرا دنباله فراخوانی شامل قوانین استفاده از رابط کاربردی برنامه‌نویسی است. از آنچه ای که دنباله فراخوانی برای ما اهمیت دارد، دیدهایی مثل نمودار کلاس برای ما مفید نخواهد بود. بهترین انتخاب گراف جهت‌دار است. علاوه بر دنباله فراخوانی، به منظور تبدیل کد به زمینه کاربر در دید سطح بالا نام و نوع اشیاء نیز باید نمایش داده شود. علت این امر در بخش ۳ آمده است. برای تولید یک دید سطح بالا از کد در Exception Tracer، از روش‌های موجود در این دسته کمک گرفته شده است. در ادامه، ابتدا اندکی در مورد GROUM [۷]، Jungloid [۲۰] و چند ابزار مشابه دیگر توضیحاتی ارائه می‌کنیم و درنهایت مزایا و معایب آن‌ها را بررسی می‌نماییم.

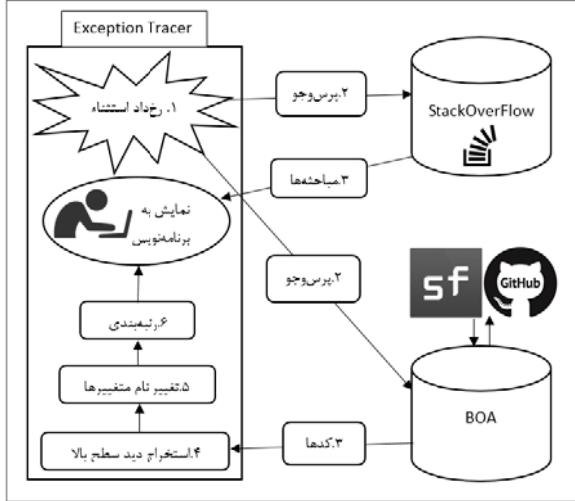
۱-۲-۲ GROUM

گراف GROUM مستقیم از کد تولید می‌شود. درخت AST کد به وسیله

همان‌طور که در مثال بالا بیان شده برای استثناءهای مربوط به عدم رعایت قوانین کتابخانه‌ها قطعه کد مفیدتر است و برای بقیه استثناءها قطعه کد مفید نخواهد بود و مباحثه‌های Stack Overflow این کمیود را جبران خواهند کرد.

انتخاب می‌شود که ممکن است اشتباه باشد [۲۲].

۳- معرفی مسئله



شکل ۲- فرآیند Exception Tracer

۴- راه کار پیشنهادی

همان‌طور که در بخش‌های قبلی عنوان شد، عدم رعایت قوانین کتابخانه‌ها و چارچوب‌های نرم‌افزاری منجر به خطأ می‌شود. دسته‌ی دیگری از خطاهای نیز به دلیل دادن ورودی غلط است. در این دسته، با اینکه برنامه‌نویس قوانین چارچوب را رعایت کرده است، اما به دلیل عدم وجود ورودی درست استثناء رخ می‌دهد. روش ارائه شده در این مقاله دو نوع یاد شده را پوشش می‌دهد. فرآیند برای ایجاد راحل، در شکل ۲ نشان داده شده است. با رخداد استثناء، پشته فراخوانی را خواهیم داشت. با استفاده از آن، متندی که فراخوانی آن باعث رخداد استثناء شده است، را می‌باییم. سپس یک جست‌وجو به وسیله موتور جست‌وجوی BOA [۲۴] به روی کدهای پروژه‌های Source Forge انجام می‌گردد تا فایل‌های شامل فراخوانی ذکر شده را پایابد. BOA یک موتور Frog جست‌وجو است که به صورت موازی به روی کدهای پروژه‌های جست‌وجو انجام می‌دهد. پرس‌وجوهای این موتور جست‌وجو، یک زبان خاص دارد که جست‌وجوی موازی را امکان‌پذیر می‌نماید. در مرحله بعد از کدهای به دست آمده گراف GROUM [۵] استخراج می‌شود. سپس از گراف‌های به دست آمده الگوهای استفاده استخراج می‌شوند. آنگاه گراف GROUM کد برنامه‌نویس استخراج می‌شود. سپس با مقایسه گراف‌ها، قطعه کدها رتبه‌بندی می‌شوند. در نهایت با استفاده از گراف‌ها، کدهای غیرمربوط از قطعه کدها حذف می‌شوند و بعد از تبدیل آن‌ها به زمینه کاربر (تغییر نام متغیرها)، قطعه کدها به برنامه‌نویس نمایش داده می‌شود. در ادامه، این گام‌ها به تفصیل بیان می‌شوند.

```

Exception in thread "main" java.io.IOException: Stream Closed
  at java.io.FileInputStream.read0(Native Method)
  at java.io.FileInputStream.read(FileInputStream.java:210)
  at Test.readFile(Test.java:20)
  at Test.main(Test.java:26)
  
```

شکل ۳- استثناء مربوط به کد شکل ۱

برنامه‌نویسان هنگام استفاده از مرورگر برای یافتن راحل استثناء خود معمولاً فقط از پیغام استثناء استفاده می‌کنند، در حالی که زمینه کد اطلاعاتی دارد که جست‌وجو را دقیق‌تر می‌کند. همچنین راحلی که برنامه‌نویس به صورت دستی پیدا می‌کند، شامل کدهای اضافی است و برای استفاده باید نام متغیرهای موجود در آن را تغییر دهد؛ این دو عمل (حذف کدهای اضافی و تغییر نام متغیر) به صورت دستی ممکن است به درستی انجام نماید و برنامه‌نویس محبو شود زمان زیادی را صرف اصلاح آن کند. علاوه بر موارد ذکر شده، تغییر محیط برنامه‌نویسی از محیط یکپارچه ایجاد نرم‌افزار به مرورگر و بالعکس عدم مرکز برنامه‌نویس را در بی خواهد داشت برای حل این مشکل روش‌های مختلفی ارائه شده است که در بخش ۲ به تفصیل بیان شدند. ولیکن، این روش‌ها نواقصی دارند که عبارتند از: (۱) ابزارها فقط نمونه کد پیشنهادی دهنده می‌باشند و یا مباحثه زمینه به خوبی استفاده نمی‌شود، (۲) نمونه کد پیشنهادی در زمینه کاربر نیست. این دلایل باعث شد تا Exception Tracer ایجاد گردد. با رخداد استثناء این ابزار فعالیت خود را آغاز می‌کند و به طور خودکار قطعه کدها و مباحثه‌های مرتبط را به برنامه‌نویس نمایش می‌دهد.

استثناءهای انواع مختلفی دارند که به صورت کلی، آنها را به دو دسته تقسیم می‌کنیم: (۱) استثناء‌ای که به علت عدم رعایت قوانین حاکم بر رابط کاربردی برنامه‌نویسی ایجاد شده است و (۲) بقیه استثناءهای دسته اول قطعه کد می‌تواند مفید باشد. زیرا برنامه‌نویس با دیدن توالی فراخوانی متدها و یا نحوه ساخته شدن اشیاء اشتباه خود را می‌یابد. اما کدهایی که استثناء نوع دوم را ایجاد کرده‌اند، قطعه کد نمی‌تواند مفید باشد؛ زیرا برنامه‌نویس دنباله فراخوانی مورد انتظار را رعایت کرده است ولی به عنوان مثال داده غلط به عنوان ورودی داده شده است.

برای درک بهتر این تفاوت به کد موجود در شکل ۱ توجه فرمایید. در این شکل کد ایجاد شی برای تعامل با یک فایل را مشاهده می‌نمایید. در صورتی که آدرس فایل غلط باشد، برنامه با استثناء File Not Found Exception را می‌گیرد. در این مثال نمونه کد نمی‌تواند برای برنامه‌نویس مفید باشد زیرا الگوهای استفاده در تعامل با فایل رعایت شده‌اند؛ ولی پارامتر ورودی غلط است. برای این قبیل استثناءها که قوانین استفاده از کتابخانه رعایت شده است اما داده ورودی غلط است، مباحثه‌های مرتبط از Stack Overflow را پیشنهاد می‌دهیم. در صورتی که در همین مثال آدرس فایل غلط نباشد، به دلیل آن که در خط ۱۰ فایل بسته شده و نمی‌توان روی فایل بسته شده عملیات خواندن انجام داد، خط ۱۱ منجر به استثناء IO Exception می‌شود. در این حالت قوانین استفاده از کتابخانه رعایت نشده است و نمونه کد مفید خواهد بود.

```

1 String log="config loaded";
2 Properties config = new Properties();
3 InputStream in =
4     new FileInputStream("config.properties");
5 config.load(in);
6 System.out.println(log);
7 String path = config.getProperty("path");
8 FileInputStream reader =
9     new FileInputStream(new File(path));
10 reader.close();
11 reader.read();
  
```

شکل ۱- کد خطadar

می‌شود. با استفاده از BOA فایل‌های شامل این فراخوانی را می‌یابیم. این پرس‌وحو کران بالایی از مجموعه کدهای مرتبط با استثناء را برمی‌گرداند، زیرا فایل‌های خروجی فقط از این منظر محدود می‌شوند که شامل فراخوانی از کد برنامه‌نویس است که استثناء در آنجا رخ داده است. در شکل ۳ استثناء مربوط به کد شکل ۱ نمایش داده شده است. در این شکل فراخوانی موردنتظر با یک اشاره‌گر نشان داده شده است.

۲-۴- استخراج کد

پرس‌وحوی ساخته شده در مرحله ۱-۴-۴ در BOA ثبت می‌شود. نتیجه این پرس‌وحو فایل‌هایی است که شامل این فراخوانی هستند. خروجی این جست‌وحو شامل لیستی از فایل‌های شامل فراخوانی موردنتظر است. لیست نهایی به ازای هر فایل خروجی شامل موارد زیر است:

- ۱- نوع مخزن نرم‌افزاری (GIT, SVN) و یا CVS.
- ۲- لینک مخزن نرم‌افزاری پروژه شامل فایل موردنتظر.
- ۳- آدرس فایل در آن مخزن نرم‌افزاری.
- ۴- شماره نسخه فایل.

با اضافه کردن آدرس فایل به لینک مخزن نرم‌افزاری بهنهایی، لینک نهایی فایل ساخته نمی‌شود. برای تولید لینک فایل باید با توجه به نوع مخزن پیش‌پردازشی به روی آدرس فایل انجام بگیرد.

۳- استخراج دید سطح بالا

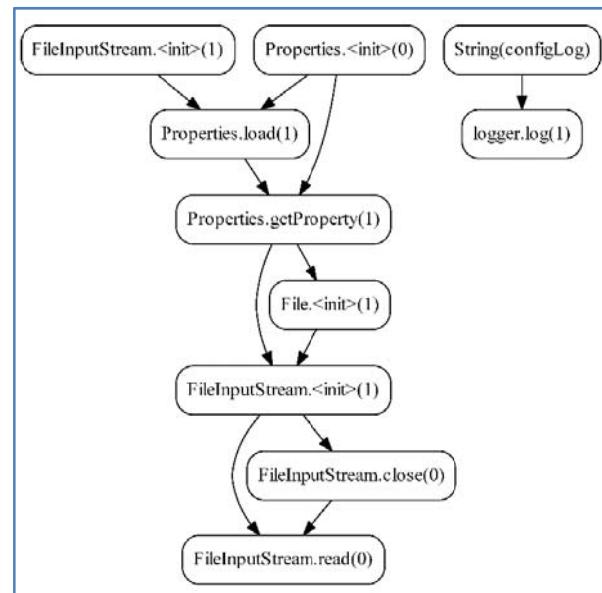
یکی از برتری‌های Exception Tracer نسبت به ابزارهای مشابه ایجاد دید سطح بالا از کد و استفاده از آن در جست‌وحوها، حذف کدهای اضافی و تبدیل به زمینه کد برنامه‌نویس است. برای ایجاد دید سطح بالا از GROUM کمک گرفته است. با ذکر یک مثال علت این انتخاب را بیان می‌کنیم. فرض کنید محلی از کد برنامه‌نویس که مربوط به فراخوانی یک متند است و استثناء در آنجا رخ داده است، مربوط به فراخوانی یک متند با نوع خروجی void باشد. در روش‌های به‌غیر از GROUM این فراخوانی‌ها در گراف ظاهر نمی‌شوند و روش ما با مشکل روبرو می‌شود. شکل ۴ گراف GROUM مربوط به کد شکل ۱ را نمایش می‌دهد (این گراف به وسیله ابزار Via Graph ایجاد شده است). GROUM از سه نوع رأس داده‌ای (مستطیل)، فراخوانی (مستطیل با گوشش‌های نرم) و شرطی (لوژی) تشکیل شده است. یال‌های GROUM همه یکسان هستند و با خط تیره نمایش داده می‌شوند. این یال‌ها براساس وابستگی داده‌ای بین رأس‌ها کشیده می‌شوند. در بخش ۴-۴-۴ نحوه مقایسه گراف‌ها را بیان می‌کنیم.

۴- رتبه‌بندی قطعه کدها

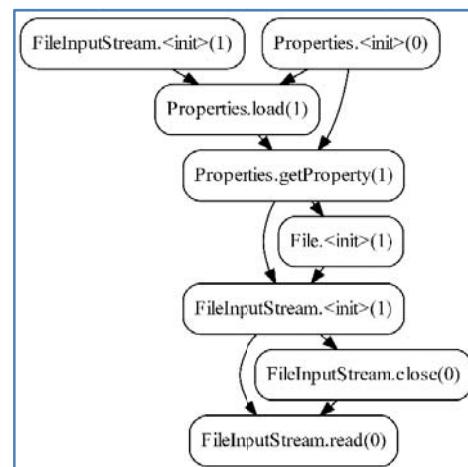
گراف شکل ۴ از دو بخش همبند تشکیل شده است. بخش‌های جدا بدین معنی است که بین رأس‌های این دو بخش هیچ‌گونه ارتباط داده‌ای وجود ندارد. در مرحله ساخت پرس‌وحو، آن فراخوانی که منجر به استثناء شده بود، شناسایی شد. از گراف فقط زیر بخش همبندی که شامل آن رأس است را نگه می‌داریم و بقیه گراف را حذف می‌کنیم. علاوه بر این کار بخش‌هایی از کد که نسبت به محل استثناء هیچ‌گونه رابطه داده‌ای (با هر عمقه) ندارند، حذف می‌شوند. در شکل ۵ گراف بعد از این عملیات قابل مشاهده است. در این شکل دو رأس مربوط به خطوط ۱ و ۶ به دلیل عدم وجود وابستگی داده‌ای مستقیم یا غیرمستقیم با رأس فراخوانی اولیه حذف شده‌اند.

۱-۴- ساخت پرس‌وحو

با استفاده از پشته فراخوانی استثناء، پرس‌وحو به زبان BOA تولید می‌شود. BOA یک موتور جست‌وحو است که به صورت موازی به روی کدهای پروژه‌های مخازن نرم‌افزاری (به صورت آزمایشی اکنون فقط به روی پروژه‌های Sewall BOA (Source Forge مبتنی بر زبان موازی Map Reduce پشتیبانی می‌کند. با پیمایش خط به خط پشته فراخوانی نام متند و کلاس داخلی ترین فراخوانی از کد برنامه‌نویس که منجر به استثناء شده است را می‌یابیم. اولین مکانی از پشته فراخوانی که خط بعدی آن مربوط به کد برنامه‌نویس است، اطلاعات فراخوانی موردنظر را مشخص می‌کند. از این خط نام متند و نام کلاس شی‌ای که آن متند به روی آن فراخوانی شده است را می‌یابد. در ادامه این رأس را "فراخوانی اولیه" می‌نامیم.



شکل ۴- گراف GROUM مربوط به کد شکل ۱



شکل ۵- گراف GROUM بعد از حذف رأس‌های غیرمربوط

پرس‌وحو تنها از فراخوانی اولیه یادشده ساخته می‌شود. پرس‌وحو با استفاده از اسم کلاس شی‌ای که فراخوانی روی آن منجر به استثناء شده و نام متند ساخته

برای تشخیص هم کتابخانه بودن کلاس‌ها از اسم بسته آن‌ها کمک می‌گیریم.تابع is Same Library بسته‌های دو کلاس را به دست می‌آورد. سپس تعداد نقطه‌های این پیشوندی بسته‌های دو کلاس را می‌تواند تعداد نقطه‌های مشترک دو کلاس را مشخص زیرشته را می‌شمارد. تعداد نقطه‌ها عمق بسته‌های مشترک دو کلاس را مشخص می‌کند. در نام گذاری بسته‌های کتابخانه‌ها نام چند بسته اول می‌تواند به عنوان تمایز بین کتابخانه‌ها در نظر گرفته شود. با بررسی کتابخانه‌های استاندارد Java و کتابخانه‌های Apache در نظر گرفتن نام دو بسته اولیه کتابخانه می‌تواند تمایز ایجاد نماید؛ در نتیجه اگر زیرشته مشترک بسته‌های دو کلاس بیش از ۲ نقطه داشته باشند، هم کتابخانه تشخیص داده می‌شوند.

$$\text{isSameLibrary}(C, C') = \begin{cases} 0 & \text{اگر } C \text{ و } C' \text{ در یک کتابخانه نباشند} \\ 1 & \text{اگر } C \text{ و } C' \text{ در یک کتابخانه باشند} \end{cases}$$

به عنوان مثال، دو کلاس `java.io.FileInputStream` و `java.io.File` هم کتابخانه تشخیص داده می‌شوند، زیرا بزرگ‌ترین زیرشته پیشوندی مشترکشان (یعنی `java.io`)، ۲ نقطه دارد.

بعد از هرس کردن گراف، محاسبه شباهت کدهای دریافت شده از Source Forge و کد برنامه‌نویس با هزینه کمتری محاسبه می‌شود. به عنوان مثال مسیرهای منتهی به رأس محل وقوع استثناء در کد شکل ۱ به شرح زیر است:

<code>File.<init>(1)</code>	<code>File.<init>(1)</code>
<code>FileInputStream.<init>(1)</code>	<code>FileInputStream.<init>(1)</code>
<code>FileInputStream.close(1)</code>	<code>FileInputStream.read(0)</code>
<code>FileInputStream.read(0)</code>	

شباهت گراف نمونه کد دریافتی در صورتی که عیناً هر دو مسیر را دارا باشند، عدد "۱" و اگر یک مسیر را دارا باشد، عدد ".۵" و در صورتی که هیچ کدام از مسیرها را دارا نباشد، عدد ".۰" خواهد بود. گراف‌ها بعد از هرس شدن، به کد تبدیل می‌شوند. به دلیل وجود مرحله هرس قبل از تبدیل به کد، قسمت‌هایی از کد که به فراخوانی اولیه مرتبط نیستند، حذف می‌شوند. همچنین بخش‌های غیروابسته به کتابخانه فراخوانی اولیه نیز حذف می‌شوند. بعد از محاسبه شباهت هر نمونه کد با کد برنامه‌نویس، لیست قطعه کدهای پیشنهادی به او نمایش داده می‌شود.

```

1 File tmpFile =
2   File.createTempFile("b12-app-paths", ".reg");
3 String[] cmdArgs =
4   new String[]{"regedit.exe", "/E"};
5 tmpFile.delete();
6 FileInputStream fis =
7   new FileInputStream(tmpFile);
8 fis.read(magic);
9 fis.close();

```

(الف) قطعه کد نمونه قبل از انتقال به زمینه کد برنامه‌نویس

```

1 File tmpFile =
2   File.createTempFile("b12-app-paths", ".reg");
3 String[] cmdArgs =
4   new String[]{"regedit.exe", "/E"};
5 tmpFile.delete();
6 FileInputStream reader =
7   new FileInputStream(tmpFile);
8 reader.read(magic);
9 reader.close();

```

(ب) قطعه کد نمونه بعد از انتقال به زمینه برنامه‌نویس

شکل ۷- قطعه کد نمونه قبل و بعد از انتقال به زمینه برنامه‌نویس

به منظور مقایسه دو گراف، از روش بیان شده است. تمام مسیرهای منتهی به رأس فراخوانی اولیه را می‌یابیم. عدد شباهت دو گراف درصد مسیرهای مشابه آن دو گراف است. P مسیر و N رأس‌ها را نشان می‌دهند. تابع $\text{is Equal Path}(P, P')$ در صورتی که این دو مسیر عیناً یکسان باشند، عدد یک و در غیر این صورت عدد صفر را برمی‌گرداند. دو مسیر در صورتی یکسان‌اند که از نظر تعداد رأس و همچنین برچسب‌های درون هر رأس کاملاً مساوی باشند. برچسب‌های درون هر رأس از کلاس شی و متodi که فراخوانی به روی آن صورت گرفته، ساخته می‌شود. یعنی در جست‌وجو نام شی تاثیری ندارد. اگر تعداد رئوس دو مسیر یکسان نباشد و یا یک رأس از یک گراف با رأس متناظر خود در گراف دیگر متفاوت باشد، خروجی این تابع صفر خواهد بود. این تابع به شرح زیر تعریف می‌شود:

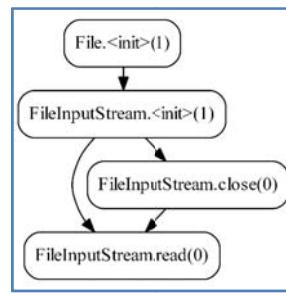
$$P = \langle N_1, N_2, \dots, N_n \rangle$$

$$\text{isEqualPath}(P, P') = \begin{cases} 1 & |P| = |P'| \wedge \forall i \in \{1, \dots, n\}: N_i = N'_i \\ 0 & \text{else} \end{cases}$$

از آنجایی که در محاسبه شباهت‌ها، همواره گراف کد برنامه‌نویس با یک گراف دیگر مقایسه می‌شود، تابع شباهت به صورت زیر محاسبه می‌شود. U گراف کد برنامه‌نویس و G گراف مربوط به نمونه کد است. رأس x همان رأس مربوط به کد محل وقوع استثناء (فراخوانی اولیه) در کد برنامه‌نویس است. همچنین، تابع paths (تمام مسیرهای منتهی به رأس n را نمایش می‌دهد.

$$\text{sim}(U, G) = \frac{\sum_{p \in \text{paths}(U, x)} \sum_{p' \in \text{paths}(G, x)} \text{isEqualPath}(p, p')}{|\text{paths}(U, x)|}$$

هر چه مسیرها طولانی‌تر باشند، هزینه محاسبه شباهت دو گراف نیز بیشتر می‌شود. در ادامه، راه حلی برای کاهش طول مسیرها بدون کم شدن دقت را با ذکر یک مثال بیان می‌کنیم. فرض کنید یک متداز کتابخانه مدنظر است که یک رشتہ به عنوان پارامتر ورودی می‌گیرد. یک کاربر ممکن است این رشتة را مستقیماً به متداز (خط اول در نمونه کد شکل ۱) و یا آن را از یک فایل بخواند و به متداز دهد (خط هفتم، در نمونه کد شکل ۱). در این مثال، در نظر گرفتن کل رأس‌های مسیر، نه تنها دقت جست‌وجو را افزایش نمی‌دهد، بلکه باعث کاهش دقت نیز می‌شود. در این مثال ما دو مسیر داریم که تابع sim آن‌ها را یکسان تشخیص نمی‌دهد، در حالی که این بخش از مسیر نباید تأثیری در نتیجه بگذارد؛ زیرا کاربر با توجه به نیازهای خود، ورودی متداز را به گونه‌ای متفاوت تعیین می‌کند. در حالت کلی، این ورودی می‌تواند هر نوع شی دیگری نیز باشد. البته اگر کلاس این شی در کتابخانه کتابخانه مربوط به رأس فراخوانی اولیه باشد، دیگر نمی‌توان با قاطعیت، رای به حذف آن از مسیر داد. بنابراین به ازای هر مسیر از انتهای آن شروع می‌کنیم، رأس‌هایی که با رأس اولیه هم کتابخانه هستند را نگه می‌داریم. مسیر از اولین رأسی که کلاسش در بسته کتابخانه نبود، قطع می‌شود. همانگونه که در شکل ۶ مشاهده می‌شود، فقط چهار رأس پایینی گراف که مربوط به کتابخانه مورد استفاده هستند، باقی می‌مانند و بقیه رأس‌ها حذف می‌شوند.

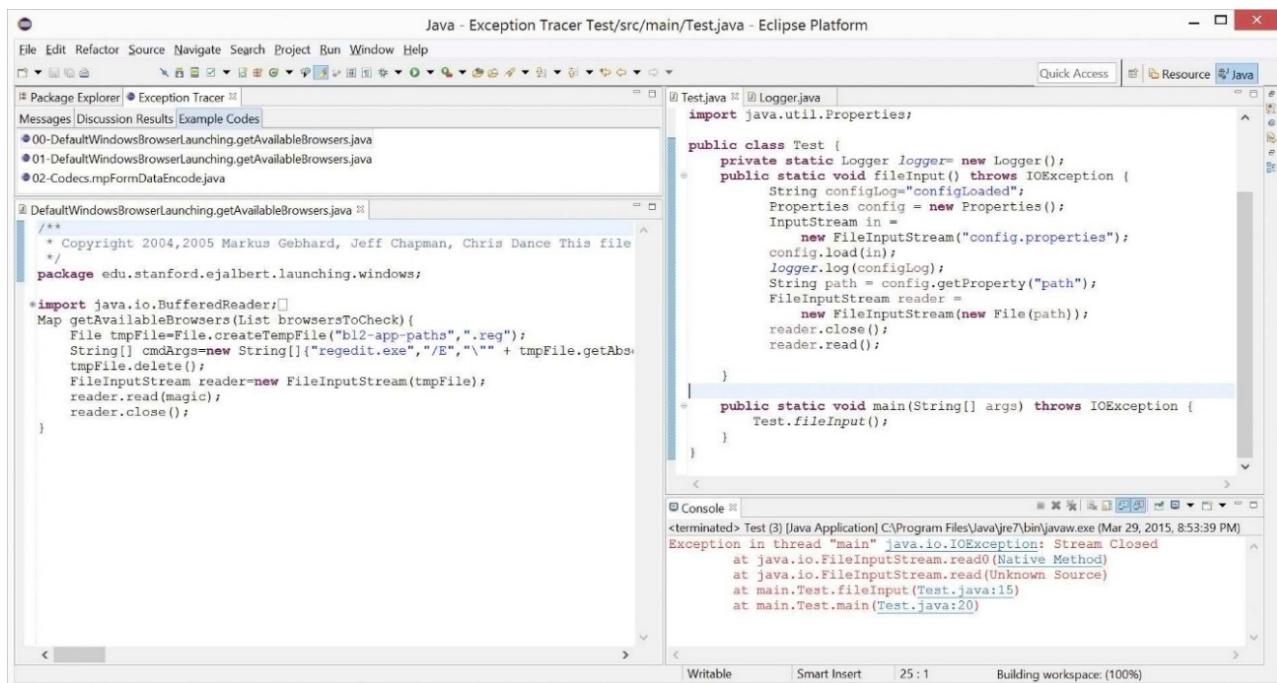


شکل ۶- گراف نمونه کد بعد از اصلاح مسیرها

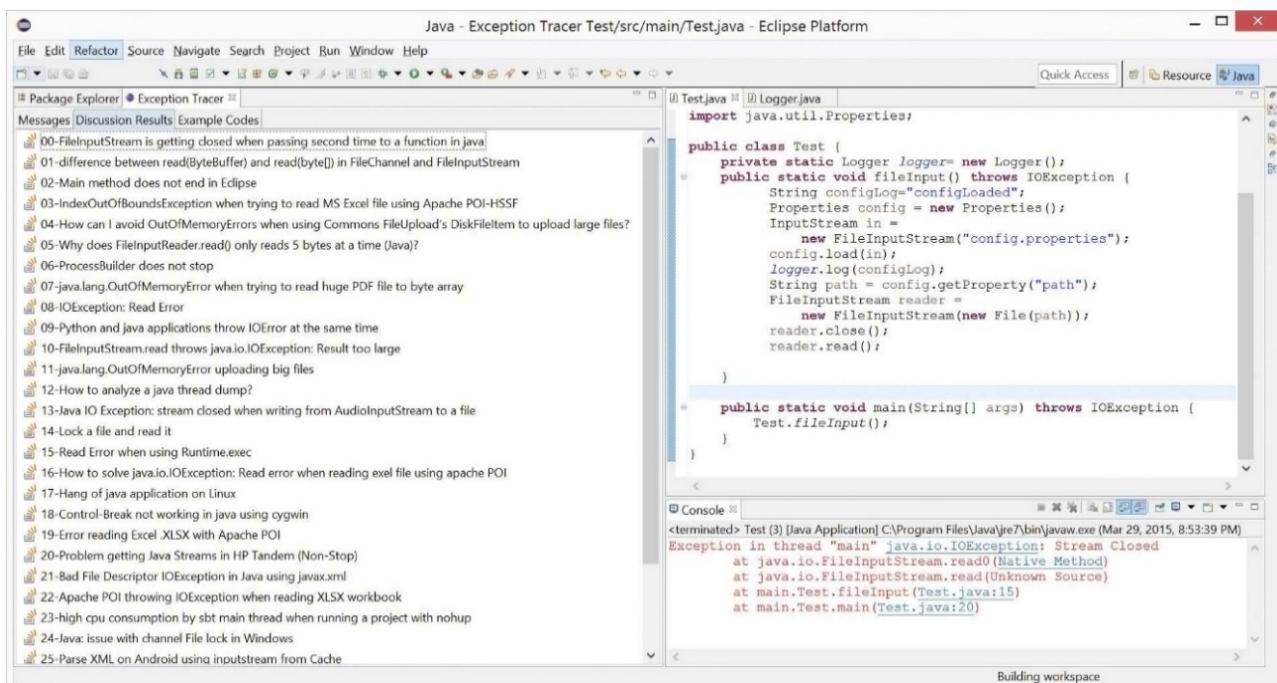
به رأس موردنظر از گراف اول بود، به عنوان رأس های متناظر مشخص می شوند. از رأس های متناظر برای تبدیل نام متغیرها استفاده می کنیم. در مثال نشان داده شده در شکل ۷، یک مثال از قطعه کد نهایی قبل و بعد از انتقال به زمینه برنامه نویس قابل مشاهده است. قبل از این تغییر، متغیر reader در قطعه کد اولیه نام دیگری داشت. همچنین، شکل ۸ محیط Exception Tracer در هنگام توصیه نمونه کد به کاربر را نمایش می دهد.

۴-۵- تبدیل به زمینه برنامه نویس

هر قطعه کد یک فضای نامی^۶ دارد که شامل اسم متغیرهای آن است. برنامه نویس هنگام استفاده از قطعه کد، باید نام متغیرها را به صورت دستی تغییر دهد که ممکن است این کار منجر به اشتباہ و در نتیجه آن، اتلاف وقت شود. با توجه به روش ارائه شده در [۲۶]، هر رأس گراف قطعه کد با هر رأس همسایه های گراف برنامه نویس مقایسه می شود. شباهت هر رأس از گراف برنامه نویس که بیشتر



شکل ۸- محیط Exception Tracer در حالت پیشنهاد نمونه کدهای مرتبط از سایت Source Forge



شکل ۹- محیط Exception Tracer در حالت پیشنهاد مباحثه های مرتبط از سایت Stack Overflow

استثناء و نحوه اندازه‌گیری بیان شده است.

۲-۵- انتخاب کتابخانه‌ها

ابزار Exception Tracer از فراخوانی محل استثناء و اطلاعات زمینه کد برنامه‌نویس به او پیشنهادهایی ارائه می‌کند. برای تولید استثناءهای مختلف از نمونه کد مربوط به کتابخانه‌های مختلفی کمک گرفته شده است.^۹ مورد از کتابخانه‌ها مربوط به کتابخانه‌های Apache هستند^۳ مورد هم مربوط به کلاس‌های پایه Java هستند. Apache در مجموع حدود ۲۷۰ پروژه (شامل کتابخانه، چارچوب و غیره) دارد و پروژه‌هایش به علت معروفیت توسعه برنامه‌نویسان زیادی استفاده می‌شود. در سه مرحله از ۲۷۰ پروژه به ۹ کتابخانه رسیدم. در مرحله اول با توجه به نام و حوزه استفاده آن پروژه، حدود ۱۰۰ کتابخانه انتخاب شدند. از بین ۱۰۰ کتابخانه با بررسی دقیق تر، ۲۰ کتابخانه‌ای که در اینترنت معروف‌تر بودند، انتخاب شدند. در حال حاضر و تا وقتی که BOA مخزن نرم‌افزاری اش را گسترش ندهد، پیش‌نیاز یافتن راه حل توسعه Exception Tracer این است که برای آن کتابخانه در Source Forge نمونه کد وجود داشته باشد. برای اطمینان از وجود نمونه کد، کتابخانه‌های معروف‌تر انتخاب شدند. در مرحله سوم سعی شد برای این کتابخانه‌ها نمونه کد ایجاد شود و با تغییراتی باعث رخداد استثناء در زمان اجرا شویم.

۳-۵- نحوه یافتن نمونه کد

به ازای هر کتابخانه یک یا چند نمونه کد صحیح از اینترنت استخراج کردیم. نمونه کدها اکثراً از وب‌سایت‌های معرفی کتابخانه‌ها و یا با جستجو در Google استخراج شدند. سپس با حذف، تغییر یا اضافه کردن یک فراخوانی، به نمونه کد دارای استثناء رسیدیم.

۴-۵- نحوه ایجاد استثناء

در بخش قبلی ما به ازای هر کتابخانه یک یا چند نمونه کد داریم که بدون مشکل اجرا می‌شوند. در این مرحله هدف این است تا با ایجاد تغییراتی در نمونه کد باعث رخداد استثناء در زمان اجرای آن بشویم. با انجام یکی از تغییرات زیر در یک یا چند خط از کد به نمونه کد آمده برای ارزیابی می‌رسیم:

- ۱- حذف فراخوانی
- ۲- اضافه کردن فراخوانی
- ۳- تغییر فراخوانی

۵-۱- اجرای ارزیابی

در این مرحله ابزار Exception Tracer را بر روی یک محیط یکپارچه ایجاد نرم‌افزار نصب نمودیم و هر یک از نمونه کدهای تولید شده در مرحله قبل را در آن اجرا کردیم. با رخداد استثناء و ارائه پیشنهاد، اعداد زیر استخراج شدند:

- ۱- تعداد نمونه کدهای استخراج شده از BOA: برای افزایش در سرعت پاسخ‌دهی ابزار، محدودیت ۲۰ تایی برای تعداد لینک‌های خروجی از BOA در نظر گرفته شده است. از آنجایی که دانلود فایل مربوط به هر لینک در حدود یک ثانیه زمان می‌برد، نمی‌توان این عدد را خیلی بزرگ در نظر گرفت. از طرفی ممکن است لینک خراب باشد، در نتیجه تعداد پیشنهادات ابزار در

۴-۶- جستجو در Stack Overflow و نمایش مباحثه‌های مرتبه

همان‌طور که در بخش‌های قبلی نیز بیان شد، درصورتی که برنامه‌نویس قوانین استفاده از کتابخانه را رعایت کرده باشد، قطعه کد لزوماً نمی‌تواند مفید باشد. همچنین، این امکان وجود دارد که قطعه کد مناسبی برای حل استثناء پیش آمده، در Source Forge موجود نباشد. در این موقع، نمایش مباحثه‌های مرتبه از Stack Overflow همراه هر جستجو، لیستی از مباحثه‌های مرتبه است. برای این منظور، با استفاده از بیان استثناء و API مورد استفاده کاربر یک پرس‌و‌جواب ساخته می‌شود و به وسیله واسط REST مهیا شده توسعه وب‌سایت Stack Overflow جستجو و انجام می‌گیرد، و نتایج جستجو در یک لیست به کاربر نمایش داده می‌شود (شکل ۹).

روش پیشنهادی در این مقاله، در ابتدا فقط شامل پیشنهاد قطعه کدهای مرتبه بود. ولیکن، در عمل متوجه شدیم که قطعه کد مرتبه به تنها یک لزوماً نمی‌تواند در همه موارد مفید باشد و در برخی مواقع مباحثه‌ها مفیدترند. به همین دلیل، به روش پیشنهادی، نمایش مباحثه‌های مرتبه نیز اضافه شد. پیشنهاد هم‌مان قطعه کد و مباحثه از نوازوری‌های این روش است. ولیکن، تمرکز اصلی ما بر ارائه قطعه کدهای مرتبه است و برای پیشنهاد مباحثه‌های مرتبه، از یک روش ساده باری جسته‌ایم. با این حال، همانطور که در بخش ارزیابی بحث خواهد شد، همین روش ساده نیز نتایج را بهبود بخشیده است.

۵- ارزیابی

ابزار Exception Tracer به منظور تسهیل فرآیند یافتن راه حل یک استثناء معروف شده است. برای ارزیابی این که چقدر این ابزار در رسیدن به این هدف موفق بوده است، در این بخش به سوالات تحقیق زیر پاسخ می‌دهیم:

- ۱- دقت ابزار برای چند نمونه کد پیشنهادی چقدر است؟
- ۲- نسبت به دیگر روش‌ها، چقدر در زمان برنامه‌نویس صرف‌جویی می‌شود؟

۵-۱- ارزیابی کارایی و دقت روش

در ارزیابی کارایی و دقت روش معیار "رتبه اولین پیشنهاد مرتبه" محاسبه شده است. بدین معنی که به ازای هر لیست خروجی که به برنامه‌نویس داده می‌شود رتبه اولین پیشنهادی که به او در یافتن راه حل کمک کرده است، چند است. به عنوان مثال، اگر به برنامه‌نویس یک لیست شامل ۱۰ قطعه کد ارائه شود و قطعه کد سوم به او در یافتن راه حل استثناء کمک کند، مقدار این معیار برای این آزمایش ۳ می‌شود. این معیار برای ۱۶ استثناء مختلف محاسبه شده است. از آنجایی که این ابزار به همراه قطعه کد، مباحثه نیز ارائه می‌کند، در هر آزمایش این معیار برای مباحثه‌ها و قطعه کدهای پیشنهادی به صورت جداگانه محاسبه شده است.

۵-۱-۱- برپایی ارزیابی

برای اندازه‌گیری دقت روش نیاز است تا چند نمونه کد که در زمان اجرا با استثناء متوقف می‌شوند، داشته باشیم و برای این نمونه کدهای خروجی ایجاد نماییم. در ادامه نحوه انتخاب نمونه کدهای دارای

۶-۱-۵- تحلیل نتایج

نتایج ارزیابی در جدول ۲ قابل مشاهده است. در ۶۲.۵ درصد موارد قطعه کد شامل راه حل استثناء در ۵ نتیجه اول وجود داشته است. در ۴۴ درصد موارد مباحثه مرتبط با استثناء در ۵ نتیجه اول نمایش داده شده، وجود داشته است. در روش ارائه شده در این مقاله تمرکز به روی تولید قطعه کد بوده است و برای جستجو در Stack Overflow Stack Overflow پرس‌وچوی بسیار ساده‌ای تولید می‌کند؛ به همین دلیل در ارزیابی انقدر ضعیف بوده است. در ۷۵ درصد موارد، قطعه کد و یا مباحثه مرتبط وجود داشته است و برنامه‌نویس با مشاهده هر دوی پیشنهادها می‌تواند به راه حل برسد. علاوه بر این معیار، می‌توان معیار MRR (Mean Reciprocal Rank) را نیز محاسبه نمود. این معیار میانگین معکوس رتبه اولین جواب است و هر چه بیشتر باشد، به معنی آن است که جواب در رتبه کمتری قراردارد و در نتیجه، Exception Tracer نشان‌دهنده بهتر بودن ابزار است. MRR برای مباحثه‌های نشان‌دهنده ۰.۴۳۲ است. این معیار برای قطعه کدها ۰.۴۳۲ است.

نهایت در حدود ۵ تا ۱۰ قطعه کد است.

-۲- رتبه قطعه کدی که شامل راه حل برطرف کردن استثناء است: برای یافتن این عدد از ابتدا تک‌تک قطعه کدها را بررسی کردیم و رتبه قطعه کدی که شامل معکوس تغییر ایجاد شده توسط ما بود را یادداشت نمودیم. به عنوان نمونه در صورتی که ما یک فراخوانی را حذف کرده باشیم، به دنبال رتبه قطعه کدی که شامل آن فراخوانی (با رعایت توالی موردنظر) بوده است، می‌گشیم.

-۳- تعداد مباحثه‌های بازگردانده شده: تعداد کل مباحثه‌هایی که ابزار به عنوان مباحثه مرتبط نمایش می‌دهد.

-۴- رتبه اولین مباحثه مرتبط با راه حل استثناء: رتبه مباحثه‌ای که در پاسخ‌های آن، راه حل رفع استثناء موجود باشد را یادداشت کردیم.

جدول ۲- نتایج ارزیابی کارایی و دقت روش

ردیف	نام کتابخانه	تعداد نتایج	Stack Overflow	مباحثه مرتبط	تعداد قطعه کدهای نتیجه شده	قطعه کد مرتب
۱	Email	۱۱		-	۱	
۲	Tar	۰		-	۱	
۳	Velocity	۴		-	۵	
۴	DB	۳۰		۳	۶	
۵	HTTP	۳۰		۹	۶	
۶	CLI	۰		-	۵	
۷	EXEC	۱۰		۱	۲	
۸	IO	۱		-	۰	
۹	file	۳۰		۲	۲	
۱۰	net	۱۷		۳	۰	
۱۱	Regular Expression	۳۰		۲	۴	
۱۲	Buffered Input Stream	۵		۲	۳	
۱۳	EMAIL-SSL	۱۲		-	۰	
۱۴	PDF	۰		-	۴	
۱۵	POI	۳۰		۱	۰	
۱۶	Velocity2	۴		-	۵	

جدول ۳- اطلاعات فردی افراد مورد مطالعه قرار گرفته شده

کاربر	مقطع	سابقه برنامه‌نویسی (سال)	میزان آشنایی با زبان جاوا	تجربه برنامه‌نویسی با زبان جاوا (سال)	میزان آشنایی با کتابخانه ها و چارچوبها	میزان آشنایی با Velocity کتابخانه	میزان آشنایی با RegEx کتابخانه
۱	کارشناسی	۲	متوسط	۲	کم	کم	کم
۲	کارشناسی	۵	کم	۱	متوسط	کم	کم
۳	کارشناسی	۴	زیاد	۲	متوسط	نا آشنا	نا آشنا
۴	کارشناسی ارشد	۵	متوسط	۴	متوسط	نا آشنا	نا آشنا
۵	کارشناسی	۴	زیاد	۴	متوسط	کم	کم
۶	کارشناسی	۳	کم	۰.۵	کم	کم	نا آشنا
۷	کارشناسی	۲	کم	۱	کم	کم	نا آشنا
۸	کارشناسی	۳	متوسط	۳	متوسط	متوسط	متوسط
۹	کارشناسی	۵	زیاد	۴	متوسط	کم	کم
۱۰	غیردانشجو	۱	کم	۰.۵	کم	نا آشنا	کم
۱۱	کارشناسی ارشد	۵	زیاد	۴	متوسط	کم	کم
۱۲	کارشناسی	۳	کم	۲	کم	نا آشنا	نا آشنا
۱۳	کارشناسی	۴	متوسط	۳	متوسط	کم	کم
۱۴	غیردانشجو	۱	کم	۰.۵	کم	نا آشنا	نا آشنا

دوره آموزش جاوا در دانشگاه و یا آموزشگاه گذرانده‌اند. ۸۵ درصد افراد انتخاب شده دانشجو بودند. در جدول ۳، اطلاعات افرادی که در این ارزیابی شرکت نمودند، نمایش داده شده‌اند.

۳-۲-۵- انتخاب نمونه کدها

نمونه کدها باید به گونه‌ای انتخاب می‌شوند که هم به راحتی با Exception Tracer بتوان به راه حل رسید و هم با استفاده از مرورگر، با این تفاسیر، نمونه کد مربوط به دو کتابخانه Velocity و RegEx انتخاب شدن. راه حل نمونه کد مربوط به RegEx در وبسایت Stack Overflow موجود است. اما کتابخانه دیگر به گونه‌ای انتخاب شده است که راه حل آن در Stack Overflow نیست و برنامه‌نویس مجبور است سایت‌های دیگر را جستجو نماید. علاوه بر این مورد نمونه کدها باید کوتاه باشند. حجیم بودن نمونه کدها باعث افزایش زمان ارزیابی و همچنین دشوار شدن ارزیابی برای افراد را در پی دارد. بیش از ۹۰ درصد افراد آشنایی خود با کتابخانه Velocity را کم و یا "تا حالا استفاده نکرده‌ام" بیان کردند. این عدد برای کتابخانه RegEx ۵۰ درصد بود.

۴-۲-۵- اجرای ارزیابی

برای مطالعه کاربران دو پرسش‌نامه طراحی شد. یک پرسش‌نامه از فرد مورد مطالعه اطلاعات اولیه را می‌خواهد و پرسش‌نامه دوم، بعد از انجام آزمایش به او داده می‌شود و جزئیات تجربه او در استفاده از ابزار را از او می‌پرسد.

۵-۲-۵- تحلیل نتایج

در این بخش تحلیل نتایج به دست آمده در دو دسته کمی و کیفی بیان می‌شود.

تحلیل کمی

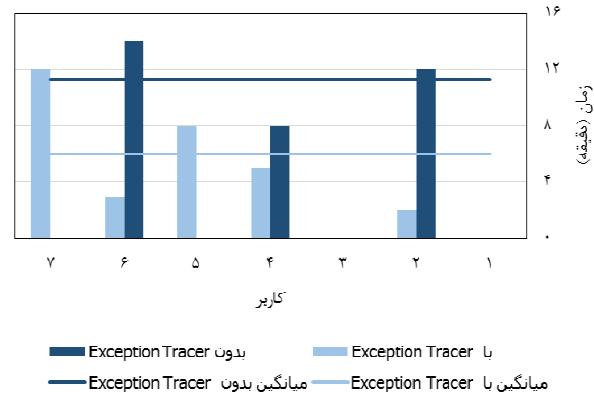
هر یک از نمونه کدهای دو کتابخانه Velocity و RegEx توسط ۱۴ نفر آزمایش شدند. هر نمونه کد توسط ۷ نفر با استفاده از Exception Tracer و توسط ۷ نفر دیگر و با استفاده از مرورگر آزمایش شدند. در مورد کتابخانه RegEx استفاده از Exception Tracer در یافتن راه حل، ۲۲ درصد صرفه‌جویی در زمان داشت. این عدد برای کتابخانه Velocity ۵۴ درصد بود. علت این امر نیز نبودن مباحثه مربوط به کتابخانه Velocity در وبسایت Stack Overflow است. نبودن مباحثه مربوط در این وبسایت باعث شد تا از ۷ نفری که باید راه حل استثنایات این کتابخانه را با مرورگر پیدا می‌کردند، فقط ۳ نفر راه حل را بیابند. این افراد با جستجو در Google نمونه کدهای مرتبط را پیدا کردند و به وسیله آن کد را اصلاح نمودند. در مورد کتابخانه RegEx همه افرادی که باید با مرورگر جستجو شوند، روش جستجو در Google و مشاهده مباحثه‌های Stack Overflow را در پیش گرفتند. در کل و بهطور میانگین این ابزار در یافتن راه حل ۳۸ درصد در زمان صرفه‌جویی می‌کند. نکته قابل توجه در این ارزیابی این است که ۳۵ درصد افراد از Eclipse استفاده نمی‌کنند، در حالی که انتظار می‌رفت این عدد کمتر باشد. نمودارهای ترسیم شده در شکل ۱۰ و شکل ۱۱، به ترتیب زمان یافتن پاسخ کد مشکل‌دار در کتابخانه‌های Velocity و RegEx را در هنگامی که از Exception Tracer استفاده شده است را با هنگامی که از آن استفاده نشده است، مقایسه می‌کند.

۲-۵- مطالعه کاربران

در بخش ۱-۵ دقت ابزار مورد بررسی قرار گرفت. در این بخش، از طریق مطالعه کاربران، میزان مفید بودن این ابزار در عمل مورد ارزیابی قرار گرفت. در این بخش به پرسش "آیا این ابزار نسبت به روش استفاده از مرورگر، برای برنامه‌نویس صرفه‌جویی زمانی دارد؟" پاسخ می‌دهیم.

۲-۵-۱- برای ارزیابی

از بین نمونه کدهایی که در زمان اجرا با استثناء متوقف شدند، دو نمونه کد انتخاب نمودیم و به ۱۴ نفر این نمونه کدها را دادیم. این افراد راه حل یکی از نمونه کدها را با استفاده از Exception Tracer یافتند و راه حل قطعه کد دیگر را با استفاده از مرورگر، در ادامه نحوه انتخاب افراد و نمونه کدها بیان می‌شود.



شکل ۱۰- نمودار مقایسه زمان‌های یافتن راه حل کد کتابخانه Velocity



شکل ۱۱- نمودار مقایسه زمان‌های یافتن راه حل کد کتابخانه RegEx

۲-۵-۲- انتخاب افراد

افرادی مورد مطالعه قرار گرفتند که آشنایی نسبتاً خوبی با زبان جاوا داشتند و در فهم و برنامه‌نویسی به این زبان مشکلی نداشتند. همچنین این افراد می‌بایست از دو کتابخانه نمونه کدها به صورت حرف‌ای استفاده نکرده باشند و برای اصلاح کد نیاز به جستجو داشته باشند. به طور میانگین، افراد انتخاب شده ۳.۵ سال تجربه برنامه‌نویسی و ۲ سال تجربه برنامه‌نویسی به زبان جاوا داشتند. همه این افراد یک

این پاسخ‌ها مشکلاتی در ابزار کشف شدند که به عنوان کارهای آتی می‌توان به آن‌ها توجه کرد. برنامه‌نویس دوم، هشتم و دوازدهم ایراد ابزار را با استفاده کاربری نامناسب مباحثه‌های Stack Overflow (فونت ریز، عملکرد نامناسب اسکرول) بیان کردند. برنامه‌نویس چهارم در مورد نمونه کدها، سرعت پایین آماده شدن قطعه کدها را مطرح کرده است. دلیل این مورد هم پاسخ‌گویی کند BOA است. در صورت بهبود ساخت افزاری BOA این مشکل رفع خواهد شد. مورد دیگر که برنامه‌نویس پنجم به آن اشاره کرده، تبع نداشتن قطعه کدهای پیشنهادی است. این مشکل زمانی پیش می‌آمد که از یک پروژه نرم‌افزاری چند قطعه کد یافته می‌شود. از آنجایی که این مثال‌ها در یک پروژه قرار دارند، سبک تمام این قطعه کدها یکسان است و برای برنامه‌نویس اطلاعات جدیدی ندارد.

مباحثه‌های پیشنهادی Exception Tracer Stack Overflow در توسعه Exception Tracer، خیلی مورد استفاده قرار نگرفت؛ علت این موضوع را می‌توان در این موارد دید: (۱) افراد در مباحثه‌ها هم به دنبال نمونه کد می‌گردند، وقتی ابزار مستقیماً نمونه کد در اختیارشان قرار می‌دهد، اول به دنبال آن می‌روند. (۲) جستجوی مباحثه‌ها خیلی دقیق نیست و با خواندن عنوان‌ین علاقه به کلیک کردن و خواندن متن مباحثه پیدا نمی‌کنند.

یکی از دو قطعه کد طوری انتخاب شده بود که راه حل آن در Stack Overflow نباشد. برای این قطعه کد، افراد کمتر بدون استفاده از ابزار به راه حل رسیدند. این مورد نشان می‌دهد که این ابزار علاوه بر پوشش استثناءهایی که جوابشان در Stack Overflow هست، برای سایر استثناءها هم می‌تواند مفید واقع شود.

از یابی ابزارهای مشابه با یک روش انجام نشده است، به همین دلیل ارزیابی Help Me Out با Exception Tracer با معیارهای گوناگون انجام شده است. ابزار با برگزاری چند کارسوق به ارزیابی ابزار خود پرداخته است. سازندگان این ابزار، دقت ابزار خود را ۵۰ درصد بیان کرده‌اند، در حالی که Exception Tracer از دقت ۷۵ درصد برخوردار است. ابزار Surf Clips معيار MRR را محاسبه کرده است که Surf Clips به معنی میانگین معکوس رتبه اولین جواب است. این معيار برای Exception Tracer مقدار ۰.۴۶ را دارد است که در مقایسه با قطعه کدهای Exception Tracer کمتر است و از آنجایی که هر چه MRR بیشتر باشد، بهتر است، ابزار Exception Tracer عملکرد بهتری دارد. البته این معيار برای مباحثه‌های Exception Tracer برابر ۰.۲۶۷ است که از Surf Clips عملکرد ضعیفتری دارد. علت این ضعف هم آن است که Exception Tracer تمرکز بیشتری روی نمونه کد داشته است و مباحثه‌هایش را با یک جستجوی ساده از سایت Stack Overflow مهیا می‌کند؛ در حالی که ابزارهای مشابه Surf Clips به‌طور تخصصی بر روی جستجوی مباحثه‌های Stack Overflow کار کرده‌اند.

۶- کارهای آتی

در بخش‌های مختلف ابزار کارهای فراوانی می‌تواند برای بهبود آن انجام شود. در زیر لیست کارهای آتی را بیان می‌کنیم:

۱- واسط کاربری مباحثه‌ها: با توجه به نظرات کیفی برنامه‌نویسان در ارزیابی انجام شده، واسط کاربری بخش مربوط به مباحثه‌ها نیاز به بهبود دارد. این بهبود به‌اندازه فونت، اسکرول و نمایش جواب تایید شده (جوابی که توسط سازنده مباحثه تایید شده است، مشخص باشد) مربوط می‌شود.

۲- تنوع در نمونه کدها: برخی اوقات پیش می‌آید که چند قطعه کد از یک پروژه نرم‌افزاری استخراج می‌شود. این قطعه کدها معمولاً یک دنباله فراخوانی یکسانی را طی می‌کنند و عملاً برای برنامه‌نویس تفاوت ملموسی ندارد. زیرا نام متغیرها نیز به فضای نام متغیرهای برنامه‌نویس منتقل شده است. از این‌رو

جدول ۴- آماره‌های آزمون

ردیف	$t_{r,i,br-et} = t_{r,i,br} - \bar{T}_{r,et}$	$t_{v,i,br-et} = t_{v,i,br} - \bar{T}_{v,et}$
۱	5.66	-
۲	-0.33	6.92
۳	0.66	-
۴	-0.33	2.92
۵	-1.33	-
۶	-1.33	8.92
۷	5.66	-

حال با استفاده از اطلاعات به دست آمده و تحلیل آماری درستی فرض "ابزار ارائه شده در این مقاله باعث صرفه‌جویی در زمان برنامه‌نویس می‌شود" را بررسی می‌کنیم.

هر نمونه کد (مربوط به کتابخانه Velocity یا RegEx) توسط ۷ نفر با ابزار Exception Tracer و توسط ۷ نفر بدون اشکال‌زدایی شده‌اند. زمانی که برنامه‌نویس آم برای یافتن راه حل نمونه کد Velocity با استفاده از Exception Tracer صرف کرده است را نشان می‌دهد. به همین شکل زمانی که برنامه‌نویس آم برای یافتن راه حل نمونه کد RegEx با استفاده از مرورگر صرف کرده است را نشان می‌دهد. میانگین زمان یافتن راه حل نمونه کد Velocity با استفاده از Exception Tracer به شرح زیر تعریف می‌شود.

$$\bar{T}_{v,et} = \frac{\sum_{i=1}^7 t_{v,i,et}}{7}$$

برای یافتن درستی فرض به ازای هر زمان عبارت $\bar{T}_{v,et} - \bar{T}_{r,et}$ را محاسبه می‌کنیم. مقادیر این متغیر تصادفی نمی‌تواند از آزمایش انجام گرفته بر روی یک نفر حاصل شود. زیرا فردی که با استفاده از ابزار ما راه حل را می‌یابد، دیگر جواب را می‌داند و همین مورد باعث می‌شود با استفاده از مرورگر سریع‌تر به جواب برسد. این اشکال در حالت برعکس نیز وجود دارد. به همین دلیل متغیر تصادفی حاصل زمان یافتن راه حل یک نفر با استفاده از ابزار منهای میانگین زمان افرادی که از مرورگر استفاده کرده‌اند، تعریف شده است. مقادیر متغیر تصادفی جدید و آماره آزمون در جدول ۴ آمده است.

از آنجایی که انتخاب افراد کاملاً تصادفی بود، با توزیع t-student فرض را بررسی نمودیم. در مورد کتابخانه Velocity از آنجایی که سه عدد بی‌نهایت داریم (به جواب نرسیده‌اند) این تحلیل را انجام نمی‌دهیم، زیرا در صورت محاسبه، به احتمال ۱۰۰ درصد فرضیه صحیح خواهد بود. فقط برای کتابخانه RegEx محاسبه می‌کنیم. میانگین و واریانس به ترتیب $S_n = ۳.۱$ و $\mu = ۱.۲۴$ هستند.

$$H_1: \mu > 0$$

$$t = \frac{\bar{T}_{r,et} - 0}{\frac{S_n}{\sqrt{n}}} = 1.06$$

$$t < t_{1-0.16,6}$$

با توجه به توزیع t-student با درجه آزادی ۶ و $\alpha = 0.16$ ، با احتمال درصد فرضیه "این ابزار نسبت به روش استفاده از مرورگر، برای برنامه‌نویس صرفه‌جویی زمانی دارد؟" درست است.

تحلیل کیفی

سؤال پایانی در پرسش‌نامه از افراد نظرشان در مورد ابزار را می‌پرسد. با توجه به

Factors in Computing Systems, pp. 1019-1028, 2010.

[4] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto , and M. Lanza, "Mining StackOverflow to turn the IDE into a self-confident programming prompter," *Proc. 11th Working Conf. on Mining Software Repositories*, pp. 102-111, 2014.

[5] M. M. Rahman, S. Yeasmin, and C. K. Roy, "Towards a context-aware IDE-based meta search engine for recommendation about programming errors and exceptions," *Proc. 21th Software Evolution Week-IEEE Conf. on Reengineering and Reverse Engineering*, pp. 194-203, 2014.

[6] P. Vekris, R. Jhala, S. Lerner, and Y. Agarwal, "Towards verifying android apps for the absence of no-sleep energy bugs," *Proc. USENIX Conf. on Power-Aware Computing and Systems*, p. 3, 2012.

[7] T. T. Nguyen, H. A. Nguyen, N. H. Pham, J. M. Al-Kofahi, and T. N. Nguyen, "Graph-based mining of multiple object usage patterns," *Proc. 7th joint meeting of the European Soft.Eng.Conf. and the ACM SIGSOFT Symp. on the Foundations of Soft.Eng.*, pp. 383-392, 2009.

[8] R. Holmes, and G. C. Murphy, "Using structural contextto recommend source code examples," *Proc. 27th IntlConf. on Soft. Eng.*, pp. 117-125, 2005.

[9] R. Holmes, R. J. Walker, and G. C. Murphy, "Strathcona example recommendation tool," *ACM SIGSOFT Soft. Eng. Notes*, vol. 30, no. 5, pp. 237-240, 2005.

[10] C. McMillan, M. Grechanik, D. Poshyvanyk, Q. Xie, and C. Fu, "Portfolio: finding relevant functions and their usage," *Proc. 33th IntlConf. on Soft. Eng.*, pp. 111-120, 2011.

[11] S. Thummalapenta, and T. Xie, "SpotWeb: Detecting framework hotspots and coldspots via mining open source code on the web," *Proc. 23th IntlConf. on Automated Soft. Eng.*, pp. 327-336, 2008.

[12] H. Cai, and R. Santelices, "Diver: precise dynamic impact analysis using dependence-based trace pruning," *Proc. 29th IntlConf. on Automated Soft. Eng.*, pp. 342-348, 2014.

[13] G. Bavota, S. Panichella, N. Tsantalis, M. D. Penta, R. Oliveto, and G. Canfora, "Recommending refactorings based on team co-maintenance patterns," *Proc. 29th IntlConf. on Automated Soft. Eng.*, pp. 337-342, 2014.

[14] M. Ghafari, C. Ghezzi, A. Mocci, and G. Tamburrelli, "Mining unit tests for code recommendation," *Proc. 22th IntlConf. on Program Comprehension*, pp. 142-145, 2014.

[15] L. Ponzanelli, A. Bacchelli, and M. Lanza, "Seahawk: stack overflow in the ide," *Proc. 35th IntlConf. on Soft. Eng.*, pp. 1295-1298, 2013.

[16] J. Yang, D. Evans, D. Bhardwaj, T. Bhat, and M. Das, "Perracotta: mining temporal API rules from imperfect traces," *Proc. 28th IntlConf. on Soft. Eng.*, pp. 282-291,

باید از هر پروژه فقط یک قطعه کد در لیست نتایج نهایی وجود داشته باشد.

- کوتاه کردن قطعه کدها: برخی متدها put ممکن است وجود داشته باشند که در قطعه کد ۵ الی ۱۰ بار پشت سر هم تکرار شده باشند. این تکرار باعث افت کیفیت قطعه کد می‌شود.

- بهبود روش یافتن متدهای استثناء شده است: برای یافتن این متدهای استثناء فراخوانی استفاده می‌کنیم. در هنگام استفاده از فریمورک‌های وب و پیچیده، اطلاعات پشت فراخوانی که هنگام رخداد استثناء نمایش داده می‌شود، خیلی پیچیده است و به همان سادگی که در بخش ۲ مطرح شد، نمی‌توان متدهای مورد نظر را یافت. بهبود این بخش از ابزار، به فرآگیرتر شدن آن کمک خواهد کرد.

- بهبود پیشنهاد مباحثه: روش پیشنهادی در این مقاله از یک روش ساده برای پیشنهاد مباحثه‌های مرتبط از Stack Overflow استفاده نموده است. صرف پیشنهاد تواماً مباحثه و قطعه کد باعث افزایش دقت کار شده است. اما در آینده می‌توان پیشنهاد مباحثه را بهبود بخشید.

۷- نتیجه‌گیری

در استفاده از کتابخانه‌ها و چارچوب‌های برنامه‌نویسی، برنامه‌نویسان قوانینی را باشیستی رعایت نمایند. عدم رعایت این قوانین ممکن است منجر به استثناء در هنگام اجرای برنامه شود. برنامه‌نویسان اغلب برای یافتن راه حل استثناء‌ها، به جستجو در سایتها پرسش و پاسخ برنامه‌نویسی و یا کدهای متن باز متولّ می‌شوند. ولیکن، این روش‌ها دارای مشکلاتی مانند عدم دقیق در ساخت پرس‌وجو و یا زمان بر بودن می‌باشند. برای کمک به حل این مشکلات، در این مقاله، ابزار Exception Tracer که بهصورتیک افزونه Eclipse پیاده‌سازی شده است، ارائه گردید. این ابزار، با استفاده از اطلاعات پشت فراخوانی و زمینه کد برنامه‌نویس، قطعه کدهایی از سایت Source Forge را که می‌توانند به حل استثناء کمک کنند، به برنامه‌نویس پیشنهاد می‌دهد. همچنین، برای کامل‌تر شدن روش پیشنهادی و کمک به برنامه‌نویسان به درک بهتر، قطعه کدهای پیشنهادی، مباحثه‌های مرتبط را نیز از سایت پرسش و پاسخ Stack Overflow نمایش می‌دهد. دقت این ابزار، با استفاده از تعدادی از نمونه کدهای مربوط به کتابخانه‌های Apache اندازه‌گیری گردید و ملاحظه شد که در ۷۵ درصد مواقع، راه حل استثناء در ۵ نتیجه ابتدایی وجود دارد. علاوه بر محاسبه دقت، میزان صرف‌جویی زمانی نسبت به جستجوی ساده در اینترنت نیز با مطالعه کاربران اندازه‌گیری شد. طبق این ارزیابی، میزان صرف‌جویی زمانی در صورت استفاده از Exception Tracer در حدود ۳۸ درصد است.

مراجع

[1] J. Cordeiro, B. Antunes, and P. Gomes, "Context-based recommendation to support problem solving in software development," *Proc. 3rd Intl Workshop on Recommendation Systems for Soft. Eng.*, pp. 85-89, 2012.

[2] Z. Gu, E. T. Barr, D. Schleck, and Z. Su, "Reusing debugging knowledge via trace-based bug search," *Proc. 27th IntlConf. on Object Oriented Programming*, pp. 927-942, 2012.

[3] B. Hartmann, D. MacDougall, J. Brandt, and S. R. Klemmer, "What would other programmers do: suggesting solutions to error messages," *Proc. 28th Conf. on Human*

عباس حیدرنوری مدرک دکترای خود در علوم کامپیوتر را در سال ۱۳۸۸ از دانشگاه واترلو کانادا و مدارک کارشناسی و کارشناسی ارشد در مهندسی نرم‌افزار خود را به ترتیب در سال‌های ۱۳۷۸ و ۱۳۸۰ از دانشگاه صنعتی شریف دریافت نموده است. بعد از تمام دوره دکترا، ایشان



به مدت یک سال به عنوان پژوهشگر پسادکترا در دانشگاه لوگانو سوییس مشغول به انجام پژوهش بوده است. ایشان سپس به مدت یک سال به عنوان مهندس ارشد Extreme Labs نرم‌افزار در زمینه تولید نرم‌افزارهای هوشمند همراه در شرکت Inc. در تورنتو کانادا مشغول به کار بوده است. دکتر حیدرنوری، از سال ۱۳۹۱ به عنوان عضو هیأت علمی در دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف مشغول به فعالیت می‌باشد. زمینه‌های پژوهشی ایشان عبارتند از مهندسی نرم‌افزار، مهندسی معکوس و بازمهندسی سیستم‌های نرم‌افزاری، و سیستم‌های توصیه‌گر در مهندسی نرم‌افزار.

آدرس پست‌الکترونیکی ایشان عبارت است از:

heydaroori@sharif.edu

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۵/۰۲/۰۸

تاریخ اصلاح: ۱۳۹۵/۰۴/۲۷

تاریخ قبول شدن: ۱۳۹۵/۰۵/۲۰

نویسنده مرتبط: دکتر عباس حیدرنوری، دانشکده مهندسی کامپیوتر،
دانشگاه صنعتی شریف، تهران، ایران.

^۱ Call Stack

^۲ Application Frame Works

^۳ Application Programming Interface (API)

^۴ Abstract Syntax Tree (AST)

^۵ Refactoring

^۶ Name space

2006.

[17] M. Gabel, and Z. Su, "Javert: fully automatic mining of general temporal properties from dynamic traces," *Proc. 16th Intl Symposium on Foundations of Soft. Eng.*, pp. 339-349, 2008.

[18] S. Sankaranarayanan, F. Ivancic, and A. Gupta, "Mining library specifications using inductive logic programming," *Proc. 30th IntlConf. on Soft. Eng.*, pp. 131-140, 2008.

[19] M. Pradel, C. Jaspan, J. Aldrich, and T. R. Gross, "Statically checking API protocol conformance with mined multi-object specifications," *Proc. 34th IntlConf. on Soft. Eng.*, pp. 925-935, 2012.

[20] D. Mandelin, L. Xu, R. Bodík, and D. Kimelman, "Jungloid mining: helping to navigate the API jungle," *ACM SIGPLAN Notices*, vol. 40, no. 6, pp. 48-61, 2005.

[21] N. Sahavechaphan, and K. Claypool, "XSnippet: mining for sample code," *ACM SIGPLAN Notices*, vol. 41, no. 10, pp. 413-430, 2006.

[22] C. Lv, W. Jiang, Y. Liu, and S. Hu, "APISynth: a new graph-based API recommender system," *Proc. 36th IntlConf. on Soft. Eng.*, pp. 596-597, 2014.

[23] S. Thummalapenta, and T. Xie, "Parseweb: a programmer assistant for reusing open source code on the web," *Proc. 22th IntlConf. on Automated Soft. Eng.*, pp. 204-213, 2007.

[24] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen, "BOA: A language and infrastructure for analyzing ultra-large-scale software repositories," *Proc. 35th IntlConf. on Soft. Eng.*, pp. 422-431, 2013.

[25] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan, "Interpreting the data: Parallel analysis with Sawzall," *Scientific Programming*, vol. 13, no. 4, pp. 277-298, 2005.

[26] A. T. Nguyen, H. A. Nguyen, T. T. Nguyen, and T. N. Nguyen, "GraPacc: a graph-based pattern-oriented, context-sensitive code completion tool," *Proc. 34th IntlConf. on Soft. Eng.*, pp. 1407-1410, 2012.

وحید امین تبار مدارک کارشناسی و کارشناسی ارشد

مهندسی نرم‌افزار خود را به ترتیب در سال‌های ۱۳۹۲ و

۱۳۹۴ از دانشگاه صنعتی شریف دریافت نموده است.

زمینه‌های پژوهشی مورد علاقه ایشان عبارتند از مهندسی

نرم‌افزار، داده‌کاوی در مخازن نرم‌افزاری، و سیستم‌های



توصیه‌گر در مهندسی نرم‌افزار.

آدرس پست‌الکترونیکی ایشان عبارت است از:

amintabar@ce.sharif.edu

آشکارسازی چهره انسان در تصاویر دیجیتال

امین‌الله مه‌آبادی

علیرضا زارعی

دانشکده فنی و مهندسی، دانشگاه شاهد، تهران، ایران

چکیده

در این مقاله روشی چالاک سریع و با دقت بالا برای تشخیص تصویر چهره انسانی در تصاویر رنگی ارایه می‌شود که بیشتر تمرکز آن در یادگیری جزئی برای کاربرد در ملیت‌های خاص و تصاویر پوستی همگون است. روش پیشنهادی بهدلیل استفاده از مدل پوست در تصاویر رنگی و الگوی چهره، متفاوت با روش‌های دیگر است و نیازی به آموزش و یادگیری به روش‌های سنتی را ندارد. مستقل از وضعیت نورپردازی، رنگ پس‌زمینه، موقعیت چهره، زاویه چرخش حول محور صورت، اندازه تصویر،وضوح تصویر و شرایط عکس‌برداری است. ضمناً تصاویر دارای ریش، عینک، با حالت خنده و بدون محدودیت تعداد چهره‌های موجود نیز آشکار می‌سازد. نتایج آزمایش‌های تصویر رنگی استاندارد و خاص حاوی چهره‌های متعدد و متفاوت انسانی با پس‌زمینه پیچیده، تحت شرایط مختلف محیطی با همزمانی وجود چندین چهره در یک تصویر، نشان می‌دهد که سیستم از دقت ۹۷٪ الی ۹۸٪ با جامعیت و مانعیت خوب در شرایط مختلف برخوردار است.

کلمات کلیدی: پردازش تصویر، کشف چهره انسان، مدل پوست، رنگ پوست.

۱- مقدمه

عدم حساسیت به تغییرات هندسی چهره است و نیازی به آموزش و یادگیری پیچیده ندارد و بیشتر تمرکز آن در یادگیری جزئی برای کاربرد در ملیت‌های خاص و تصاویر پوستی همگون است.

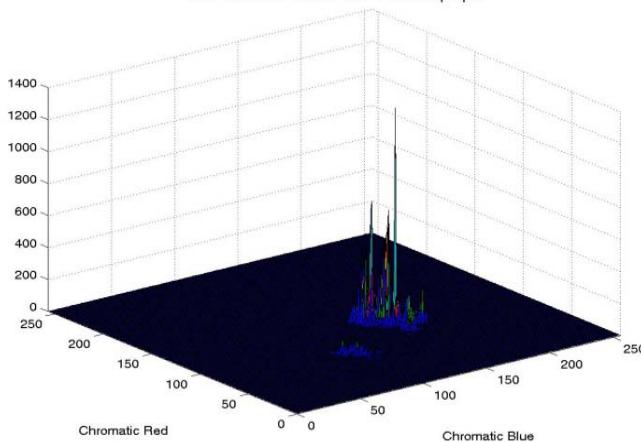
در ادامه، مقاله ابتدا به کارهای مرتبط در بخش ۲ می‌پردازد. در بخش ۳ روش پیشنهادی طرح می‌شود. مدل پوست در بخش ۴ و نحوه بخش‌بندی پوست در بخش ۵ ارایه می‌گردد. نحوه تشخیص نواحی پوستی در بخش ۶ بیان می‌شود. بخش ۷ مبنی مدل چهره است و نتایج آزمایش‌های تجربی‌تر در بخش ۸ ارایه و نهایتاً مقاله در بخش ۹ نتیجه‌گیری می‌شود.

۲- کارهای مرتبط

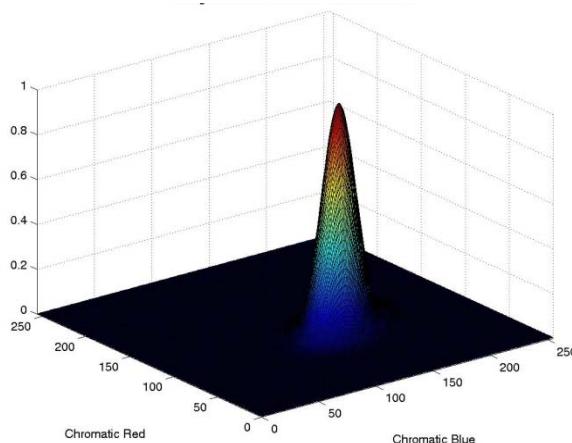
یک سیستم مطمئن تشخیص چهره براساس رنگ پوست و ویژگی‌های هندسی متتمرکز است. دقت این روش‌ها ۹۵٪ است [۲۶]. همچنین ارایه روش‌های شبکه عصبی با براساس مدل‌های آماری در فضای رنگی هستند [۲۸-۲۹]. مثلاً از هیستوگرام برای بیان بافت محلی و شکل تصویر سراسری استفاده می‌کنند [۲۶].

هدف نهایی تحقیقات بینایی ماشین ایجاد یک سیستم خودکار تشخیص چهره است که بتواند کارآتر از انسان باشد. اما بهدلیل استفاده از مدل‌ها و ساختارهای بسیار پیچیده توسط مغز انسان، رسیدن بهقدرت و سرعت تشخیص او توسط ماشین فعلاً بعید بهنظر می‌رسد [۲-۱]. تشخیص چهره بهعنوان گام اساسی در تشخیص هویت انسان، لب خوانی، بازیابی تصویر براساس محتوا، فشرده‌سازی تصویر و مانند آن است. مطالعات بسیاری درین زمینه انجام گرفته است. عموماً کشف چهره با استفاده از روش‌های مبتنی بر مدل، ویژگی‌ها، تطبیق الگو، بافت، شبکه عصبی، عمق و رنگ پوست انجام می‌شود [۳-۱۷]. اهمیت روش‌های موجود در استفاده از تصاویر خاکستری است در حالی که بیشتر تصاویر امروزی رنگی هستند [۱۱-۲۱]. روش‌های جدید از تصویر رنگی برای تشخیص چهره بهره می‌برند [۲۲-۲۵]. در این مقاله روشی جهت آشکارسازی چهره در تصاویر رنگی، مستقل از رنگ پس‌زمینه، شرایط عکس‌برداری و تعداد چهره‌های موجود در تصویر ارایه می‌شود. این روش پیشنهادی بدون محاسبات پیچیده دارای سرعت پردازش بالا.

مختلف در محدوده کوچکی از فضای رنگی کروماتیک قابل طبقه‌بندی است [۱۱]. گرچه ظاهرا بمنظور می‌رسد که رنگ پوست افراد در محدوده وسیعی متغیر باشد اما از لحاظ رنگ خیلی کمتر تفاوت دارند تا تفاوت در روشنایی. به عبارت دیگر رنگ پوست افراد مختلف بسیار بهم نزدیک هستند و تفاوت محسوس، در شدت آن است که نتایج این تحقیقات به استفاده از مدل رنگ پوست در فضای کروماتیک در روش پیشنهادی منجر شده است.



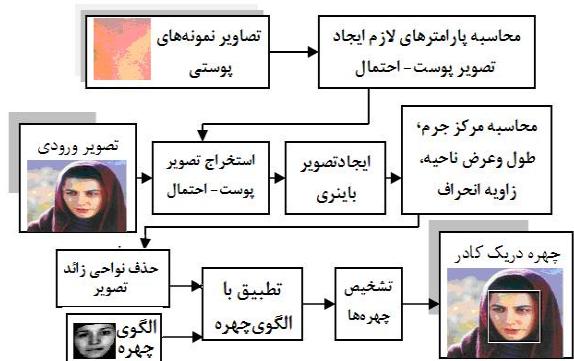
شکل ۲- توزیع رنگ پوست افراد مختلف



شکل ۳- برآش رنگ پوست در توزیع گوسی

از ۱۵۰ نمونه پوستی گرفته شده از ۳۵ تصویر رنگی برای تعیین توزیع رنگ پوست انسان در فضای کروماتیک استفاده شده و نمونه‌ها از افراد با نژادهای گوناگون آسیایی، اروپایی، قفقازی، آفریقایی و لاتین اخذ گردیده است. سپس نمونه‌های پوستی با استفاده از فیلتر پایین گذر برای کاهش نویز بهبود یافته است. شکل ۲، توزیع رنگ این نمونه‌های پوستی را در فضای کروماتیک نشان می‌دهد. هیستوگرام رنگی شکل ۲ نشان‌دهنده توزیع رنگ پوست افراد مختلف در فضای کروماتیک است که در محدوده مشخصی گرد آمده است. توزیع رنگ پوست با مدل گوسی ($N(m, c)$) و به شرح رابطه (۱) نمایش داده می‌شود. شکل ۳، توزیع گوسی برآش شده مطابق داده‌ها را ارایه می‌دهد. با بکارگیری این مدل رنگ پوست و با استفاده از رابطه (۲)، احتمال تعلق هر پیکسل به ناحیه پوستی در تصویر بدست می‌آید. مدل رنگ پوست می‌تواند تصویر رنگی را به یک تصویر خاکستری بدل کند که در آن ارزش هر پیکسل نمایانگر احتمال پوست بودن آن نقطه است. با آستانه‌گیری تطبیقی از این تصویر، یک تصویر باینری بدست می‌آید

از ماشین بردار پشتیبان و کلاس‌بندی آدابوست با ساختار آموزش در فضای مختلف نیز استفاده شده است [۲۸]. کارآیی هیستوگرام و مدل‌های تشخیص پوست و یافتن مدل‌های هیستوگرام بسیار عالی عمل می‌کنند و باعث افزایش دقت و کاهش هزینه می‌شوند [۳۰]. مدل‌های تشخیص براساس رنگ پوست بسیار ساده، قابل اعتماد و سیار سریع تر از روش‌های دیگر هستند. ما روشی براساس رنگ پوست، فضای احتمال پوست و الگوی چهره ارایه می‌دهیم که از ویژگی‌های سادگی، قابلیت اعتماد، سرعت، تشخیص در پس‌زمینه پیچیده، وجود چهره با اندازه‌های متفاوت و تحت محیط‌های مختلف برخوردار است.



شکل ۱- الگوریتم آشکارسازی چهره انسان در تصاویر رنگی

۳- روش پیشنهادی

روش پیشنهادی ما در قالب یک سیستم تعیین هویت پیاده‌سازی شده است. روال انجام کار مطابق الگوریتم شکل ۱ است که در ابتدا با دریافت نمونه‌های پوستی واستخراج مدل پوست-احتمال آغاز می‌شود؛ محاسبات مرکز جرم، ابعاد ناحیه و زاویه انحراف انجام می‌گیرد؛ نواحی زائد حذف می‌گردد؛ با الگوی چهره (تهیه شده از نمونه‌ها) تطبیق داده می‌شود؛ نهایتاً چهره‌های انسانی استخراج می‌گردد. روش ما می‌تواند به عنوان یک مرحله چالاک و کم وزن برای روش‌های دیگر استفاده شود که بیشتر تمرکز ما در مرحله یادگیری جزئی برای کاربرد در ملیت‌های خاص و تصاویر پوستی همگون است.

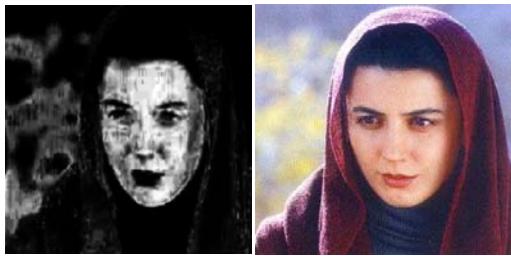
۴- مدل پوست

به منظور بخش‌بندی نواحی پوستی برمنای رنگ، به یک مدل رنگ پوست با قابلیت تطبیق برای افراد با رنگ‌های مختلف پوستی و شرایط نوری متفاوت نیاز داریم. در این بخش یک مدل رنگ پوست در فضای رنگی کروماتیک برای بخش‌بندی نواحی پوستی ارایه می‌شود. نمایش RGB متدالون عکس‌های رنگی جهت توصیف رنگ پوست مناسب نیست زیرا در این فضا سه مولفه سازنده آن (G, B, R) نه تنها نماینده رنگ‌ها بلکه شاخص عنصر دیگری به نام روشنایی نیز می‌باشند. روشنایی می‌تواند در عکس‌های متفاوت از چهره یک فرد نیز متفاوت باشد که این مسئله ممکن است از تفاوت نور محیط ناشی شود. لذا روشنایی مولفه قابل اعتمادی در زمینه جداسازی نواحی پوستی و غیرپوستی نیست. برای حل این مشکل می‌توان از نمایش عکس در فضای کروماتیک استفاده کرد بدليل آنکه در این فضا می‌توان روشنایی را از مولفه‌های رنگی متمایز نمود.

رنگ‌های کروماتیک به شکل گستره‌های برای بخش‌بندی تصاویر بکار برده می‌شوند که در روش پیشنهادی نیز به نحو مطلوبی به بخش‌بندی تصاویر به نواحی پوستی و غیرپوستی کمک می‌کند. تحقیقات نشان داده که توزیع رنگ پوست افراد

۱-۶- حفره

برای محاسبه تعداد حفره‌های یک ناحیه، از عدد اول آن ناحیه مطابق رابطه (۳) استفاده شده است. در آن رابطه، E عدد اول، C نشان‌دهنده بخش‌های بهم متصل و H نماینده تعداد حفره‌های نواحی متصل است. در روش پیشنهادی تعداد نواحی متصل با عدد یک مقداردهی می‌شود زیرا در هر مرحله با یک ناحیه سروکار داریم پس رابطه (۴) را خواهیم داشت. اگر سیستم تعداد حفره‌ها را بیشتر از یک تشخیص دهد، مرحله تطبیق الگو انجام می‌گیرد و درغیراینصورت ناحیه مورد نظر از ادامه فرآیند حذف می‌گردد.



(الف) تصویررنگی اصلی (ب) تصویرپوست-احتمال

شکل ۴- نمونه یک تصویر رنگی و تصویر پوست-احتمال



(الف) تصویر بخش‌بندی شده (ب) تصویرپوست-احتمال

شکل ۵- تصویر بخش‌بندی شده انسانی و تصویر پوست-احتمال



(الف) ناحیه پوستی (ب) ناحیه بخش‌بندی شده

شکل ۶- ناحیه پوستی مشخص شده و نواحی بخش‌بندی شده

$$E = C - H \quad (3)$$

$$H = 1 - E \quad (4)$$

۲- مرکز جرم

برای مطالعه هر ناحیه در ابتدا باید مرکز آن ناحیه بدست آید که روش مناسب آن، محاسبه مرکز جرم ناحیه است. زیرا در تصاویر باینری مرکز جرم، همان مرکز

که ناحیه پوستی و غیرپوستی در آن متمایز شده‌اند. هدف این مرحله یادگیری برای افزایش دقت آن در کاربرد ملیت‌های خاص و تصاویر پوستی همگون است.

$$\text{mean: } m = E(x) \quad x = (r, b)^T \quad T: \text{Transpose} \\ \text{covariance: } C = E[(x - m)(x - m)^T] \quad (1)$$

$$P(r, b) = \exp \left[-0.5(x - m)^T C^{-1}(x - m) \right] \quad (2) \\ \text{where: } x = (r, b)^T$$

۵- بخش‌بندی پوستی

تصویر پوست-احتمال یک تصویر خاکستری است که در آن مقدار سطح خاکستری هر پیکسل، نمایانگر احتمال تعلق آن به ناحیه پوستی می‌باشد. یک نمونه تصویر رنگی و تصویر پوست-احتمال نشان‌دهنده آن در شکل ۴ نشان داده شده است و در آن تصویر ناحیه پوستی روشن‌تر دیده می‌شود. از آنجا که نواحی کاندیدای پوست نسبت به نواحی دیگر روشن‌تر است لذا نواحی پوستی را می‌توان با آستانه‌گیری از تصویر پوست-احتمال جدا کرد.

برای پردازش تصاویر افراد مختلف با رنگ‌های پوستی متفاوت و از نژادهای متمایز، استفاده از یک آستانه ثابت، مناسب و مقدور نیست. برای دستیابی به یک آستانه بهینه در هر تصویر نیاز به آستانه‌گیری تطبیقی است. در آستانه‌گیری تطبیقی کاهاش مرحله‌ای مقدار آستانه، مستقیماً باعث رشد تعداد نواحی کشف شده می‌گردد. هر چند این افزایش در تعداد نواحی به تدریج رو به کاهاش می‌گذارد زیرا درصد نواحی کشف شده به ۱۰۰٪ می‌کند. ولی این نواحی کشف شده در صورت کوچک بودن مقدار آستانه به سرعت رشد خواهد کرد به گونه‌ای که حتی نواحی غیرپوستی را نیز در بر خواهد گرفت. مقدار آستانه‌ای که در آن کمترین میزان رشد نواحی کشف شده را شاهد باشیم به عنوان مقدار آستانه بهینه تلقی می‌گردد.

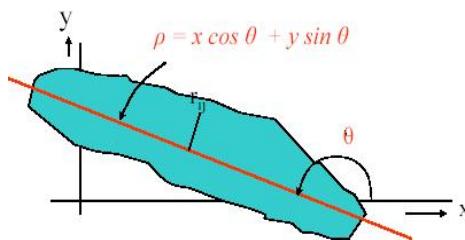
در شکل ۵ تصویر بخش‌بندی شده مشاهده می‌گردد. ملاحظه می‌شود همه نواحی کشف شده حاوی چهره انسانی نیستند. حتی برخی از نواحی کشف شده نواحی غیرپوستی هستند که به اشتباہ در بین نواحی پوستی قرار گرفته‌اند. این امر بهدلیل شباهت رنگ این نواحی به نواحی پوستی است. لذا در مراحل بعدی، جستجوگر سیستم از برخی ویژگی‌های چهره برای تعیین محل چهره در تمام مناطق هر رنگ پوست استفاده می‌کند.

۶- نواحی پوستی

یک ناحیه پوستی تصویر، ناحیه بسته‌ای است که دارای ۰ یا ۱ حفره باشد. مرز این ناحیه با پیکسل‌هایی با ارزش ۱ مشخص می‌شود. ضمن آنکه می‌توان یک ناحیه پوستی را به صورت بخش‌های بهم متصل در نظر گرفت. حفره‌ها در یک تصویر باینری، پیکسل‌هایی با مقدار صفر هستند. برای تعیین تعداد نواحی از روش برچسب‌گذاری (۸ همسایگی) استفاده شده است. برای هر پیکسل در صورتی که حداقل یک همسایگی آن دارای برچسب باشد آن پیکسل را به همان همسایگی نسبت داده شده و در غیراینصورت از یک برچسب جدید استفاده می‌شود. در پایان برچسب‌ها شمرده شده و تعداد نواحی تعیین می‌گردد. برای جداسازی هر بخش، از برچسب آن منطقه استفاده و بعد از یافتن آن بخش بقیه نواحی صفر می‌شود. سپس تعیین می‌گردد که آیا ناحیه کشف شده موردنظر نماینده یک چهره هست یا خیر (شکل ۶).

۲۶ تصویر مختلف تمام رخ زن و مرد بدون ریش ایجاد شده و بعنوان الگوی شناخت چهره از آن بهره‌برداری می‌گردد. البته اگر نمونه‌ها افزایش یابد می‌توان برای هر ملتی الگوی خاص چهره آن را یافت و دقت تشخیص بیشتر می‌شود. در این بخش چگونگی تطبیق تصاویر نواحی متناظر با ناحیه پوستی و تصویر الگو انجام می‌گیرد. برای بخش‌های متناظر با ناحیه پوستی در ابتدا حفره‌های موجود در تصویر پر شده سپس تصویر حاصل در تصویر اصلی ضرب می‌شود (شکل ۹).

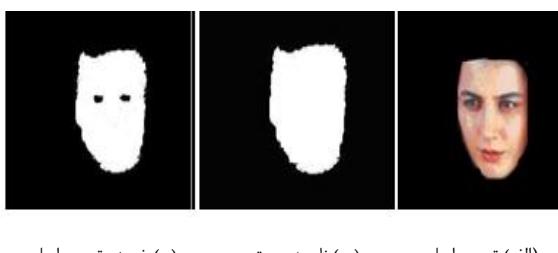
الگوی چهره باید کاملاً برروی ناحیه موردنظر قرار داده شود. برای این منظور باید آن را براساس طول و عرض محاسبه شده در قسمت‌های قبلی، تغییر اندازه داد. الگوی چهره باید به اندازه زاویه انحراف ناحیه چرخش پیدا کند و نواحی اضافی آن حذف شود (شکل ۱۰). سپس مرکز چهره الگوی، تغییریافته محاسبه می‌شود و در نهایت تصویر سیاه و سفیدی که تصویر تغییریافته الگو بر روی آن قرار داده شده به دست می‌آید. ضریب همبستگی دو بعدی r دو ماتریس تصویر $A_{m \times n}$ و $B_{m \times n}$ مطابق رابطه (۷) محاسبه می‌گردد. با استفاده از نتایج تجربی حاصل از تصاویر مختلف، مقدار آستانه پایین برای این ضریب جهت تصمیم‌گیری نهایی چهره بودن ناحیه $\theta = 0.58$ تعیین شد. در صورت پاسخ مثبت، سیستم با یک کادر ناحیه مورد نظر را مشخص می‌کند. نمونه تصویر خروجی با چهره‌های بسیار در شکل ۱۱ مشاهده می‌شود.



شکل ۷- زاویه چرخش چهره در تصویر



شکل ۸- الگوی نهایی چهره انسان



شکل ۹- مراحل تشخیص ناحیه پوستی

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2 \right) \left(\sum_m \sum_n (B_{mn} - \bar{B})^2 \right)}} \quad (7)$$

ناحیه است. مرکز جرم (\bar{x}, \bar{y}) این نواحی به شرح رابطه (۵) محاسبه می‌شود که در آن B یک ماتریس $m \times n$ نشان‌دهنده ناحیه و A تعداد پیکسل‌های خاموش آن ناحیه است.

$$\bar{x} = \frac{1}{A} \sum_{i=1}^n \sum_{j=1}^m j B[i, j] \quad \bar{y} = \frac{1}{A} \sum_{i=1}^n \sum_{j=1}^m i B[i, j] \quad (5)$$

۶-۳- تطبیق جهت

از آنجاکه برخی چهره‌های تصویر دارای زاویه انحراف می‌باشند، برای دستیابی به نتیجه بهتر باید تصویر الگو مطابق جهت چهره بچرخد لذا نیاز به محاسبه زاویه انحراف چهره است. برای این کار از گشتاور مرکزی مرتبه دوم استفاده شده و زاویه خط گذرنده از ناحیه را که دارای گمرتین مجتمع مربعات فاصله از نقاط ناحیه با محورها باشد، اندازه گیری می‌کند. به این ترتیب زاویه انحراف θ (شکل ۷) به شرح رابطه (۶) محاسبه می‌شود.

۶-۴- ابعاد ناحیه

برای تعییر اندازه تصویر الگو و تطابق با ابعاد تصویر مورد جستجو جهت تحمل عدم پوشش ناحیه گردن و نواحی زیر گردن در تصویر نیاز به محاسبه طول و عرض ناحیه است. برای این کار ابتدا حفره‌های ناحیه موردنظر پر شده، سپس با استفاده از زاویه انحراف آن را به حالت کاملاً عمودی درآورده و با استفاده از حرکت چهار اشاره‌گر در چهار جهت (بالا، پایین، چپ و راست)، مرازهای ناحیه با برخورد به پیکسل‌های روشن، یافت شده و طول و عرض محاسبه می‌شود. برای بهمود عملیات تصمیم‌گیری از تقسیم ارتفاع به عرض ناحیه استفاده می‌شود. $\text{Ratio} = (\text{Height} / \text{Width})$. این نسبت برای انسان‌ها، عددی در حدود ۱ است.

$$\begin{aligned} c &= \sum_{i=1}^n \sum_{j=1}^m (y'_{ij})^2 B[i, j] \\ \theta &= 1/2 \arctan \frac{b}{a-c} \\ a &= \sum_{i=1}^n \sum_{j=1}^m (x'_{ij})^2 B[i, j] \\ b &= 2 \sum_{i=1}^n \sum_{j=1}^m x'_{ij} y'_{ij} B[i, j] \\ x' &= x - \bar{x} \quad y' = y - \bar{y} \end{aligned} \quad (6)$$

برای کم کردن تلفات بهطور تجربی یک حد بالا و یک حد پایین برای این نسبت انتخاب شده است. نتایج تجربی نشان داد که مقدار بهینه حد پایین 0.6 و مقدار بهینه حد بالا در حدود 1.2 است. مواردی وجود دارد که ناحیه موردنظر نماینده یک ناحیه چهره‌ای باشد اما این نسبت از 1.2 بالاتر است. این مسئله در موقعی است که بخش‌هایی از گردن و زیر آن پوشیده نباشد. در این موارد به تجربه می‌توان نسبت را از 1.2 فرض و نواحی خارج از این نسبت را از تصویر حذف نمود.

۷- مدل چهره

مهم‌ترین مرحله این روش مرحله تطبیق الگوی چهره با ناحیه کاندیدا جهت تصمیم‌گیری نهایی است. تصویر الگویی که در شکل ۸ ملاحظه می‌شود از میانگین

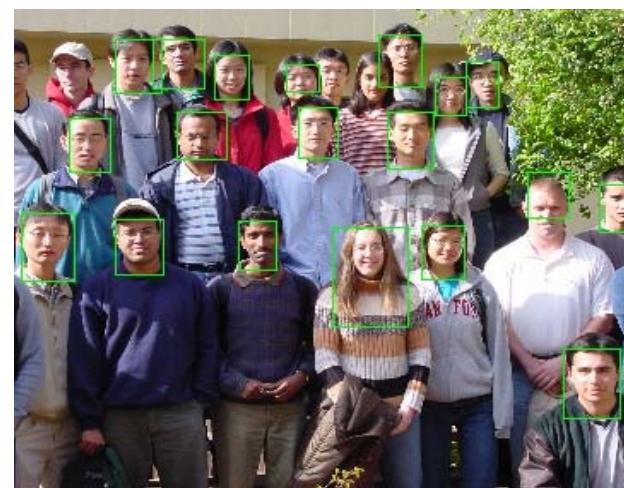
یادگیری جزئی برای کاربرد ملیت‌های خاص و تصاویر پوستی همگون است. نتایج آزمایش‌ها بر روی نمونه‌ها استاندارد و نمونه‌های ایرانی اخذ شده در این بخش بیان می‌گردد. در ابتدای آزمون پارامترهای مورد نیاز سیستم در تصاویر اندازه‌گیری شده و سپس نتایج تشخیص چهره اخذ گردیده است.



شکل ۱۰- (الف) الگوی چرخش‌یافته (ب) حذف زوائد

ما برای آزمون از دو پایگاه داده‌ای تصاویر خانوادگی ایرانی حاوی ۶۰۰ تصویر رنگی چند چهره‌ای و دارای تصاویر خاص با ابعاد و جهت متفاوت چهره بهره برده‌ایم. برای ملیت‌های متفاوت از تصاویر استاندارد دارای ۴۰۰ تصویر رنگی با داشتن چند چهره هم‌زمان در یک تصویر بهره‌برداری شده است. سیستم تشخیص چهره پیشنهادی با استفاده از MATLABR2014b بر روی ماشین 2.4 GHZIntel Windows 7 تحت 4 GB RAM با ساختار Corei4 و داشتن حافظه ۴ تخت ۷ پیاده‌سازی آزمون شده است.

۱-۸- تنظیمات آزمایش‌ها



شکل ۱۱- نتیجه تشخیص با تصویر دارای ۲۲ چهره

جدول ۱- نمونه‌های آزمایشات بر روی تصاویر برای اخذ پارامترها

تعداد تصاویر کشفی غیرچهره	درصد چهره کشف شده	اندازه ضریب همبستگی	اندازه نسبت طول به عرض ناحیه	نمره
۴	۸۰	۰.۵۸	۰.۲	۱
۳	۸۰	۰.۵۸	۰.۴	۲
۲	۸۰	۰.۵۸	۰.۶	۳
۲	۷۵	۰.۵۸	۰.۸	۴
۱	۸۰	۰.۵۸	۱.۰	۵

جدول ۲- نمونه‌های آزمایشات بر روی تصاویر برای اخذ پارامترها

تعداد تصاویر کشفی غیرچهره	درصد چهره کشف شده	اندازه ضریب همبستگی	اندازه نسبت طول به عرض ناحیه	نمره
۵	۷۵	۰.۵۸	۰.۲	۱
۵	۷۵	۰.۵۸	۰.۴	۲
۳	۷۵	۰.۵۸	۰.۶	۳
۲	۶۸	۰.۵۸	۰.۸	۴
۱	۶۰	۰.۵۸	۱.۰	۵

۸- نتایج آزمایش‌های تجربی

با استفاده از نتایج تجربی آزمایش‌ها، بهترین مقدار کمینه برای ضریب همبستگی ۰.۵۸ بدست آمد. ضمن آن که مقدار بهینه برای نسبت طول و عرض ناحیه چهره‌ای در محدوده ۰.۶ تا ۱.۲ در نظر گرفته شده است. نتایج جداول ۱ و ۲ حاصل از آزمایش بر روی ۲۰۰ تصویر شامل ۳۰۰ چهره است. با استفاده از این نتایج، بدینه است که مقدار بهینه کمینه برای نسبت طول به عرض ناحیه ۰.۶ است زیرا بالاترین نرخ تشخیص درست و کمترین نرخ تشخیص نادرست در این مقدار رخ می‌دهد. با انجام آزمایش‌های مشابه مقدار بهینه ضریب همبستگی ۰.۵۸ یافت شد. با استفاده از نتایج و انجام آزمایش بر روی ۲۰۰ تصویر مختلف، درصد چهره‌های کشف شده ۸۹٪ بدست آمده که نشان دهنده دقت سیستم است که برای تصاویر دارای ریش تاحدودی دچار افت می‌شود. در مقابل در شرایط کنترل شده پس زمینه و وضعیت چهره، دقت رشد ۹۸٪ را نشان می‌دهد. سرعت سیستم

با توجه به آن که سعی این مقاله در ارایه یک روش سبک و چالاک برای تشخیص چهره انسان است و توجه به ارایه روشی با دقت و صحت مناسب دارد لذا از مقایسه با نتایج دیگران خودداری شده است. روش‌های نوین بیشتر براساس روش‌های پیچیده و بسیار دقیق تشخیص هستند ولی روش ما می‌تواند به عنوان یک مرحله چالاک و کم وزن برای روش‌های دیگر استفاده شود که بیشتر تمرکز آن در مرحله

and Machine Intelligence (PAMI), vol. 24, no. 1, pp. 34-58, 2002.

[4] V. Bruce, and A. W. Young, "Understanding Face Recognition," Br. J. Psychol., no. 77, pp. 305-327, 1986.

[5] R. Chellappa, C. Wilson, and S. Sirohey, "Complex Background Using Color Information and SGLD Matrices," Proc. First Int'l Workshop Automatic Face and Gesture Recognition, pp. 238-242, 1995.

[6] J. L. Crowley, and J. M. Bedrune, "Integration and Control of Reactive Visual Processes," Proc. Third European Conf. Computer Vision, no. 2, pp. 47-58, 1994.

[7] A. J. Calder, and et. al., "Configural Information in Facial Expression Perception," J. Exp. Psychol. Hum. Percept. Perform., vol. 26, pp. 527-551, 2000.

[8] H. Wang, and S. F. Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video," IEEE Trans. Circuits and Systems for Video Technology, vol. 7, no. 4, pp. 615-628, 1997.

[9] J. Cai, A. Goshtasby, and C. Yu, "Detecting Human Faces in Color Images," Image and Vision Computing, vol. 18, no. 1, pp. 63-75, 1999.

[10] A. J. Calder, and A. W. Young, "Understanding the Recognition of Facial Identity and Facial Expression," Nature Rev. Neurosci., vol. 6, no. 8, pp. 641-651, 2005.

[11] Y. Gong, and M. Sakauchi, "Detection of Regions Matching Specified Chromatic Features," Computer Vision and Image Understanding, vol. 61, no. 2, pp. 263-269, 1995.

[12] A. J. Colmenarez, and T. S. Huang, "Face Detection with Information-based Maximum Discrimination," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 782-787, 1997.

[13] J. L. Crowley, and F. Berard, "Multi-modal Tracking of Faces for Video Communications," Proc. Computer Vision and Pattern Recognition, Puerto Rico, pp. 640-645, June 1997.

[14] T. F. Cootes, and C. J. Taylor, "Locating Faces using Statistical Feature Detectors," Proc. Second International Conference on Automatic Face and Gesture Recognition, pp. 204-209, 1996.

[15] T. K. Leung, M. C. Burl, and P. Perona, "Finding Faces in Cluttered Scenes using Random Labeled Graph Matching," Proc. Fifth International Conference on Computer Vision, pp. 637-644, 1995.

[16] H. A. Rowley, S. Bluja, and T. Kanade, "Neural Network-based Face Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, 1998.

[17] K.-K. Sung, and T. Poggio, "Example-based Learning for View-based Human Face Detection," IEEE Trans. Pattern

در مقایسه با سیستم‌های مشابه که از الگوریتم‌های شبکه عصبی استفاده می‌کنند بیشتر است.

۴-۸- نتایج تجربی سیستم

می‌توان دقت بالا با بازیافت بسیار پایین به دست آورد که افزایش یکی با کاهش دیگری همراه است. معیار F ترکیبی از معیار دقت و بازیافت است که قدرت کار با مجموعه‌های غیر متعادل را نشان می‌دهد. در جدول ۳ مشخص است که نتایج روش ارایه شده از نظر دقت و فراخوانی بسیار خوب است و معیار F در جدول نشان از جامعیت نسبتاً خوب روش ارایه شده است. آزمایش‌های مختلف ما نشان داد که هرچه تعداد تصاویر بالاتر رود تشخیص درست چهره کاهش می‌یابد. البته توансیم این مساله را با الگوی چهره کنترل کنیم و اخذ نمونه‌های پوستی به کاهش آن بسیار کمک کرده است. بنا داریم که در آینده با ترکیب یادگیری محلی این حساسیت را بسیار کاهش دهیم.

جدول ۳- نتایج تشخیص چهره در تصاویر در شرایط مختلف

معیار	شرایط کنترل شده		شرایط کنترل نشده	
	تک چهره	چند چهره	تک چهره	چند چهره
Precision	.۹۸	.۹۸۶	.۹۷۵	.۹۷۸
Recall	.۵۶	.۵۷	.۵۶	.۵۶
F-Score	.۷۱۱	.۷۲	.۷۱۱	.۷۱۲

۹- نتیجه‌گیری

روش پیشنهادی می‌تواند به عنوان یک الگوریتم صنعتی سریع و دقیق تشخیص خودکار چهره انسان تحت شرایط کنترل شده مانند مراکز پژوهشی قانونی و شرایط کنترل نشده به منظور تعیین هویت افراد به کار رود. به نحو موثری چهره‌های موجود در تصاویر رنگی را نشان می‌دهد و دارای دقت بالا است. در مواردی که تصویر در شرایطی نزدیک به شرایط عکس گذرنامه باشد دارای دقت ۹۸٪ است. در مواردی که پس زمینه تصویری دارای رنگ مشابه با نواحی پوششی باشد نیز موفق عمل می‌کند. در صورت استخراج الگوی چهره برای تصاویر نزدیک به هم (مانند یک ملیت خاص) درصد خطای سیستم کمتر می‌شود. برای کشف چهره‌های نیم‌رخ کافی است الگوی چهره‌ای برای چهره نیم رخ ایجاد و در مرحله تطبیق الگو از آن استفاده شود. برای آینده سعی بر آن داریم که با بکارگیری یادگیری محلی کمی از حساسیت روش پیشنهادی بکاهیم.

مراجع

[1] P. Sinha, and et. al., "Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About," Proceedings of the IEEE, vol. 94, no. 11, pp. 1948-1962, 2006.

[2] K. Lander, and L. Chuang, "Why are Moving Faces Easier to Recognize," Visual Cognition, vol. 12, pp. 429-442, 2005.

[3] M. H. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," IEEE Trans. Pattern Analysis

[30] A. Goswami, and A. Datar, "Frontal Enhanced Face Detection using Skin Pixels and ANN," International Journal of Current Engineering and Technology, vol. 6, no. 1, 2016.

عیرضا زارعی لیسانس خود را در رشته مهندسی کامپیوتر نرم افزار از دانشگاه پیام نور شیراز و فوق لیسانس خود را در رشته معماری کامپیوتر از دانشگاه شاهد اخذ نموده است. تحقیقات مورد علاقه نامبرده پردازش تصویر و بینایی ماشین است.



آدرس پست الکترونیکی ایشان عبارت است از:

a.zarei1600@gmail.com

امین‌الله مه‌آبادی تحصیلات خود را در رشته مهندسی برق سخت‌افزار و معماری کامپیوتر به انجام رسانده و اکنون استادیار گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه شاهد است. تحقیقات مورد علاقه نامبرده پردازش تصویر، حمل و نقل هوشمند و شبیه‌سازی هوشمند است.



آدرس پست الکترونیکی ایشان عبارت است از:

mahabadi@shahed.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۵/۰۵/۱۱

تاریخ اصلاح: ۱۳۹۵/۰۵/۳۱

تاریخ قبول شدن: ۱۳۹۵/۰۶/۲۵

نویسنده مرتبط: دکتر امین‌الله مه‌آبادی، دانشکده فنی مهندسی، دانشگاه شاهد، تهران، ایران.

^۱ Precision

^۲ Recall

^۳ F-Score

Analysis and Machine Intelligence, vol. 20, no. 1, pp. 39–51, 1998.

[18] Y. Dai, and Y. Nakano, "Face-texture Model based on SGLD and its Applications in Face Detection in a Color Scene," Pattern Recognition, vol. 29, no. 6, pp. 1007–1017, 1996.

[19] L. Xu, and et. al., "Segmentation of Skin Cancer Images," Image and Vision Computing, vol. 17, no. 1, pp. 65–74, 1999.

[20] K. C. Yow, and R. Cipolla, "Finding Initial Estimation of Human face location," Proceedings of the Second Asian Conference on Computer Vision, Singapore, no. 3, pp. 514–518, 1995.

[21] K. C. Yow, and R. Cipolla, "A Probabilistic Framework for Perceptual Grouping of Features for Human Face Detection," Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Vermont, USA, 1996.

[22] K. C. Yow, and R. Cipolla, "Feature-based Human Face Detection," Image and Vision Computing, no. 15, pp. 713–735, 1997.

[23] Q. Chen, H. Wu, and M. Yachida, "Face Detection by Fuzzy Pattern Matching," Proceedings of the Fifth International Conference on Computer Vision, pp. 591–596, 1995.

[24] Y. Miyake, and et. al., "Facial Pattern Detection and Color Correction from Television Picture and Newspaper Printing," Journal of Imaging Technology, vol. 16, no. 5, pp. 165–169, 1990.

[25] K. Sobottka, and I. Pitas, "Segmentation and Tracking of Faces in Color Images," Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp. 236–241, 1996.

[26] A. Verma, S. A. Raj, A. Midya, and J. Chakraborty, "Face Detection using Skin Color Modeling and Geometric Feature," 3rd International Conference on Informatics, Electronics & Vision (ICIEV), pp. 1-6, 2014.

[27] F. Alizadeh, S. Nalousi, and C. Savari, "Face Detection in Color Images using Color Features of Skin," World Academy of Science Engineering and Technology, vol. 5, no. 4, pp. 366-372, 2011.

[28] K. N. Lu, and A. N. Venetsanopoulos, "Face Recognition using LDA-based Algorithms," IEEE Neural Networks Transaction, vol. 14, no. 1, pp. 195–200, 2013.

[29] T. Talele, S. Kadam, and A. Tikare, "Efficient Face Detection using Adaboost," IJCA Proc. on International Conference in Computational Intelligence (ICCIA'12), 10, 2010.

افزایش گذردهی در شبکه‌های موردي به وسیله مدل صفت‌چابکی با در نظر داشتن احتمالات اختصاص اولیه

ابوالفضل دیانت^۱احمد خونساری^۲*محمد امین کشتکار^۱^۱دانشکده برق و کامپیوتر، دانشگاه تهران، تهران، ایران^۲*پژوهشکده علوم کامپیوتر، پژوهشگاه دانش‌های بنیادی (IPM)، تهران، ایران

چکیده

شبکه‌های موردي بی‌سیم از دیرباز مورد توجه محققان و صاحبان صنعت بوده است. با گذشت زمان این شبکه‌ها حوزه‌های متفاوتی از صنایع و کاربردها را در برگرفته است و نوع خاصی از این شبکه‌ها با نام شبکه‌های اقتضایی بین خودرویی وارد صنعت خودرو شده است. در این مقاله به مدل‌سازی تاخیر و دستیابی به گذردهی بالاتر در شبکه‌های اقتضایی بین خودرویی، به کمک بهره‌مندی از فن آوری‌های مختلف دسترسی بی‌سیم می‌پردازیم. برای تحقیق این امر، مدل‌سازی و محاسبه تاخیر را به کمک یکی از مدل‌های تئوری صفت باتام "چابکی" انجام داده‌ایم و برای کاهش زمان تأخیر ارسال بسته‌ها روش جدیدی ارائه گردیده که در آن خودروها از چند تکنولوژی دسترسی بی‌سیم به صورت همزمان بهره می‌برند و بدین ترتیب بسته‌های داده بین چند صفحه که هر کدام به یک رابط ارتباطی بی‌سیم متصل است پخش می‌شوند و سپس به خودروهای متفاوتی مقصده فرستاده می‌شوند. این بسته‌ها با توجه به اینکه با چه احتمالی در ابتدا به صفحه‌ای کوتاه‌تر ارسال اختصاص یابند، منجر به تاخیرهای متفاوتی می‌گردد که در مقاله به آن پرداخته شده است. این مدل‌سازی نهایتاً موجب کاهش مدت زمان انتظار بسته‌های داده در صفحه‌ها می‌شود. پس از حل مدل ریاضی مسئله، که در آن حالت توزیع ایستا نیز محاسبه گردیده، نتایج حاصل از شیوه‌سازی، که تکنیک چندرادیویی با مدل صفت مذکور را با کارهای پیشین مقایسه می‌کند، نشان‌دهنده کاهش چشمگیر زمان تأخیر ارسال بسته‌ها و افزایش گذردهی کلی شبکه است.

کلمات کلیدی: شبکه اقتضایی بی‌سیم، شبکه اقتضایی بین خودرو، مدل صفت‌چابکی، چند رادیویی.

۱- مقدمه

است که در آن خودروها به عنوان گره‌های سیار مورداستفاده قرار گرفته و امکان برقراری ارتباط بین خودروها و ایستگاه‌های کنار جاده‌ای (RSU)^۳ و همچنین بین خود خودروها را فراهم می‌سازد. [۱۴] [۱۳] ساختار شبکه‌های VANET دارای ویژگی‌های ذاتی است که آن را از سایر شبکه‌ها متفاوت می‌سازد: برقراری ارتباطات با سرعت بالا بین خودروها و فراهم کنندگان سرویس در هر زمان و کمبود رابطه‌ای ارتباطی یک دست بین تولیدکنندگان خودروها. برای رفع این نیازمندی‌ها، تلاش‌های زیادی در انجمن ارتباطات موسسه مهندسان برق و الکترونیک^۴ برای تعیین و تعریف مجموعه‌ای از پروتکل‌های استاندارد، معماری، سرویس‌ها و رابطه‌ای ارتباطی صورت گرفته که مجتمع‌منجر به برقراری ارتباط امن بی‌سیم بین خودرو-خودرو (V2V)^۵ و بین زیرساخت-خودر (V2I)^۶ که با نام

شبکه‌های موردي سال‌هast که مورد توجه پژوهشگران در زمینه‌های مختلف است و حوزه‌های کاربردی مختلفی را نیز در بر گرفته است. این شبکه‌ها شامل مجموعه‌های از گره‌های توزیع شده‌اند که با هم‌دیگر به طور بی‌سیم ارتباط دارند. گره‌ها می‌توانند کامپیوتر میزبان با مسیریاب باشند، به طور مستقیم بدون هیچگونه نقطه دسترسی با هم‌دیگر ارتباط برقرار کنند، سازمان ثابتی ندارند و بنابراین در یک توپولوژی دلخواه شکل گرفته‌اند. این شبکه‌ها دارای دو نوع نسبتاً مجزا با نام شبکه‌های حسگر بی‌سیم و شبکه‌های اقتضایی سیار می‌باشد. شبکه‌های اقتضایی بین خودرویی (VANETs)^۱ نوعی از شبکه‌های اقتضایی سیار (MANETs)^۲

۲- کارهای مرتبط

۱-۲- پیش‌زمینه

نوع ارتباطات موجود در شبکه‌های VANET به دلیل ماهیت خاص این شبکه که قبل اشاره شد قابل تقسیم به چهار دسته زیر می‌باشد. [۱۱]

ارتباطات خودرو- خودرو (V2V): در این نوع ارتباط، خودروها تلاش می‌کنند تا بتوانند با استفاده از رابطه‌های ارتباطی خود یک شبکه نظیری‌به‌نظیر را تشکیل دهند و به روبدل اطلاعات بپردازند. رابطه‌های ارتباطی موردادستفاده در این نوع عموماً مبتنی بر DSRC و یا ۸۰۲.۱۱a/b/g/n می‌باشد. [۱۲]

ارتباطات خودرو- ایستگاه کنار جاده‌ای (V2R): این دسته از شبکه اقتضایی شامل خودروهای متحرک و ایستگاه‌های ارتباطی کنار جاده‌ای می‌شود. DSRC، WiMAX، ۸۰۲.۱۱a/b/g/n تکنولوژی‌های موردادستفاده در این نوع شامل

ارتباطات خودرو- زیرساخت (V2I): خودروها می‌توانند با زیرساخت‌های

ارتباطی موجود برای پوشش طیف وسیع تری از برد رادیویی ارتباط برقرار کنند. تکنولوژی موردادستفاده در این نوع ارتباطی عموماً از نوع شبکه‌های سلولی (LTE, LTE Advance) می‌باشد. [۱۳]

دیگر انواع ارتباطات: در برخی منابع پژوهشگران نوع دیگری از ارتباطات را نیز مدنظر قرار داده‌اند. مانند ارتباط بین خودرو با دستگاه‌های شامل پنهانی باند ابری^{۱۱} که در آن خودرو با یک مرکز داده ارتباط برقرار می‌کند و به ارسال و دریافت اطلاعات می‌پردازد. [۱۰]

امروزه کمپانی‌های سازنده خودرو قادر به استفاده از فناوری‌های دسترسی بی‌سیم متفاوتی بر روی خودروها هستند. این تکنولوژی‌ها با هدف افزایش ایمنی جاده‌ها، بهره‌وری ترافیک و فراهم آوردن آرامش رانندگان و مسافران به وسیله برنامه‌های کاربردی ایمنی و غیرایمنی موردادستفاده قرار می‌گیرند. بعضی از این تکنولوژی‌ها بر یک ساختار مرکزی برای ارتباطات بین خودروها تکیه دارد و در مقابل برخی دیگر به صورت اقتضایی (همکاری توزیع شده) عمل می‌کنند. در زیر مختصراً لیستی از فناوری‌های اصلی دسترسی بی‌سیم که در شبکه‌های VANET قابل استفاده هستند بیان می‌گردد. [۱۷] [۱۸]

سیستم‌های سلولی (2/2.5/2.75/3G): نکته اصلی در مورد سیستم‌های سلولی استفاده مجدد از فرکانس‌های محدود در دسترس، برای ارائه سرویس می‌باشد. ارتباطات سیستم جهانی سیار (GSM) یکی از استانداردهای سیستم سلولی است که ماکریم ترخ تبادل داده ۵Gbps را به دست می‌دهد. نسل دوم و سوم و چهارم این تکنولوژی با نام‌های ۲G, 3G, 4G پیز هم‌اکنون با ترخ ارسال داده بالاتر قابل استفاده است. [۴]

پروتکل ارتباطی بی‌سیم (WLAN/WiFi(802.1a/b/g/n)): دسترسی بی‌سیم به صورت WiFi ارتباطات از نوع V2V و V2I را ممکن می‌سازد. استاندارد ۸۰۲.۱۱a در باند 5GHz و با نرخ داده ۵۴Mbps و طول برد ۳۸ متر برای ارتباطات داخلی و ۱۴۰ متر برای ارتباطات بیرونی قابل دسترسی است. سایر پیشرفت‌ها در استانداردهای بعدی نظیر IEEE80.211b/g/n موجب بالاتر رفتن نرخ ارسال داده و بهبود مسائل امنیتی است.

پروتکل ارتباطی بی‌سیم WiMAX: این پروتکل که در سال ۲۰۰۴ توسط مؤسسه IEEE پذیرفته شد و با عنوان استاندارد IEEE80.2.16 می‌شود پوشش رادیویی وسیع تری (تا ۱۵ کیلومتر) با نرخ ارسال داده بلا و قابل اطمینان به کاربران ارائه می‌دهد.

دسترسی بی‌سیم در محیط‌های خودرویی (WAVE)^۷ شناخته می‌شود شده است. این نوع ارتباط بر پایه استانداردهای IEEE 802.11p و IEEE 1609 IEEE می‌باشد. [۱۲]

با توجه به افزایش روزافزون حجم اطلاعات و محبوبیت دستگاه‌های ارتباطی سیار، تقاضا برای افزایش میزان پهنای باند از سوی کاربران افزایش یافته و توجه بسیاری از محققان بر روی انتقال حداکثری جریان اطلاعات با زمان تأخیر کم و نسبت تحويل بالا از طریق شبکه‌های VANET متتمرکز گردیده است. برای دست‌یابی به این هدف بسیاری از پژوهشگران و توسعه‌دهندگان سیستم راهکارهایی مانند بهمنه‌سازی در لایه کنترل دسترسی میانی [۵]، بهره‌وری بهتر از پروتکل‌های مسیریابی کنونی و ارائه پروتکل‌های مسیریابی جدید، بهره‌وری از پروتکل ارتباطات برد کوتاه اختصاصی (DSRC)^۸ در کنار کانال‌های خالی ایستگاه‌های تلویزیونی [۳]، کاهش ازدحام و برخورد در شبکه‌های VANET و ... ارائه داده‌اند که هر کدام موجب بهتر شدن برخی از پارامترهای کارایی شبکه و همچنین پیدایش دلن مشکلات جدیدی شده است. بسیاری از این راهکارها مناسب سناریوها و حالات خاصی از شبکه‌های VANET بوده و بهطور کلی فراغیر نمی‌باشند. [۱۹]

هدف ما در این مقاله ارائه یک روش چند رادیویی جدید برای افزایش گذردهی در مسیریابی شبکه‌های اقتضایی بین خودرویی هست که از تکنولوژی‌های ارتباطی بی‌سیم مختلف که بر روی خودروها وجود دارد استفاده کرده و در هر حالت بهترین رابط ارتباطی را انتخاب کند و اطلاعات خود را به RSU‌ها و یا سایر خودروها منتقل نماید.

از آنجایی که در دنیای ارتباطات کنونی تکنولوژی‌های دسترسی بی‌سیم مختلفی در دسترس هست، می‌توان با ترکیب این تکنولوژی‌ها در یک معماری مناسب و منطبق کردن آن با نیازمندی‌های شبکه‌های VANET از این قابلیت جهت کاهش زمان تأخیر در انتقال اطلاعات استفاده کرد. شایان ذکر است نیازی نیست همه این تکنولوژی‌های مختلف توسط کمپانی‌های تولید کننده خودرو بر روی خودرو نصب شود و می‌توان از دستگاه‌های موبایل و یا سایر دستگاه‌های ارتباطی سرنوشتی و رانندگان خودروها به عنوان رابطه‌های ارتباطی بی‌سیم در مواقعي که در دسترس هستند بهره جست.

در روش پیشنهادی این مقاله علاوه بر تکنولوژی DSRC که براساس استاندارد ۸۰۲.۱۱p IEEE 1609 IEEE بسیار مرسوم است از سایر تکنولوژی‌ها نظیر شبکه مخابراتی (LTE^۹), WiMAX و موارد دیگری که در قسمت "پیش‌زمینه" بدان‌ها اشاره خواهد شد، برای برقراری ارتباطات و انتقال داده‌ها استفاده می‌شود. یک خودروی موجود در شبکه به رابطه‌های ارتباطی شامل DSRC, WiMAX, LTE مجهر شده و می‌تواند از طریق هر کدام از آن‌ها به تنها و یا به صورت همزمان داده‌های خود را ارسال نماید. در زمان ارسال خودرو می‌تواند بسته‌های داده را به هر کدام از این رابطه‌ای بی‌سیم ارتباطی جهت ارسال تحويل نماید. بنابراین بسته‌های داده در صفحه‌ای جدایی از هر کدام از آن‌ها پیشنهادی می‌گیرند. در روش پیشنهادی ما از مدل صف "چابکی"^{۱۰} استفاده می‌کنیم تا بتوانیم میزان تأخیر بسته‌ها در صف را کاهش دهیم و بنابراین نرخ ارسال داده در شبکه افزایش بابد.

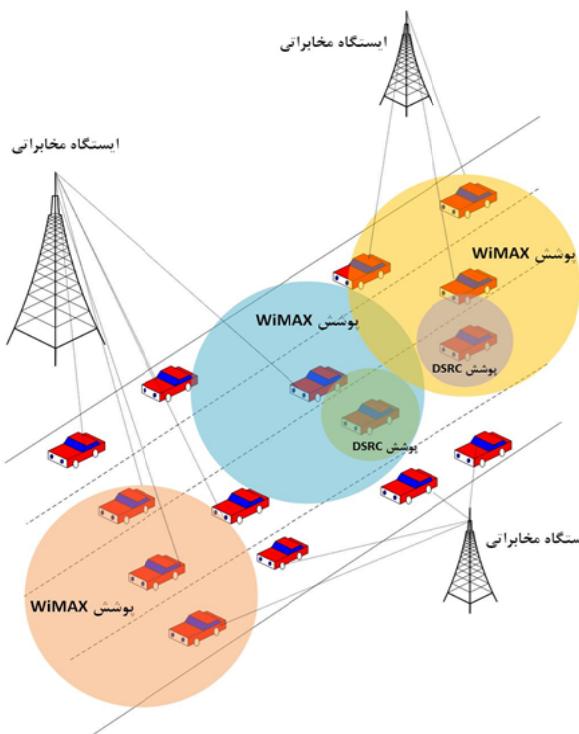
در ادامه مقاله در قسمت بعد به بیان انواع ارتباطات و تکنولوژی‌های بی‌سیم موجود در شبکه‌های VANET می‌پردازیم. در قسمت "پیش‌زمینه" و مدل صف چابکی "علاوه بر مرور کارهای مشابه قبلی، به معرفی و نحوه مدل کردن مسئله چند رادیویی با استفاده از مدل صف چابکی می‌پردازیم. در قسمت ۳ روش شبیه‌سازی و ارزیابی مسئله بیان شده و نهایتاً در قسمت ۵ نتایج حاصل از روش را بیان می‌کنیم.

زمان تأخیر کاهش و میانگین نرخ تحویل بسته‌ها افزایش یابد. شمایی از شبکه VANET موردنظر را در شکل ۱ مشاهده می‌کنید.

۲-۳- مدل صفحه‌چابکی

در هر سیستمی که تعداد دستگاه‌های پردازنده از تعداد بسته‌ها بیشتر باشد با مدل‌های صفحه‌چابکی مرتبط به آن مواجه هستیم. به طور کلی در چنین سیستمی یکی از چهار مورد زیر در مورد رفتار بسته‌ها اتفاق می‌افتد: [۱]

- بسته‌ها تک به تک منتظر می‌مانند و طول صفحه‌چابکی می‌کنند تا سرویس پیغایند.
- بسته‌ها به دلیل طولانی بودن صفحه‌چابکی منتظر صفحه‌چابکی می‌کنند.^{۱۴}
- بسته‌ها پس از مدتی انتظار در صفحه‌چابکی منتظر می‌کنند.^{۱۵}
- بسته‌ها یک صفحه‌چابکی را ترک کرده و به صفحه‌چابکی دیگر می‌پیوندند.^{۱۶}



شکل ۱- معماری روش چند رادیویی پیشنهادی

در سایر کارهای قبلی که در VANET برای پخش اطلاعات در حالتی که خودروها با بیش از یک رابط ارتباطی بی‌سیم مجهز بوده‌اند انجام شده، یا به طور کلی از رابطه‌های مختلف برای انتقال نوع داده‌های متفاوت ارسال می‌شده و بسته‌ها اجازه رفتن از یک صفحه‌چابکی به صفحه‌چابکی دیگر را نداشته‌اند. مانند [۲] که از رابطه ارتباطی DSRC که همیشه در دسترس است برای ارسال بسته‌های اورژانسی و از رابطه ارتباطی DSRC که همیشه در دسترس است برای ارسال بسته‌های غیراورژانسی بهره برده است، و یا در سایر مقالات از لحاظ معماري و امکان‌سنگي موردن بحث قرار داده‌اند و به بررسی پارامترهای کارایی شبکه نپرداخته‌اند. [۷] [۸] در این مقاله ما از مدل صفحه‌چابکی برای فرستادن بسته‌های داده، خواه اورژانسی و خواه غیراورژانسی، استفاده می‌کیم. بدین ترتیب برای هر خودرویی که قصد ارسال اطلاعات را داشته باشد سه صفحه‌چابکی را در نظر می‌گیریم که منتظر پردازش (ارسال شدن) هستند (شکل ۲). در حالت معمولی خواه طول یک صفحه‌چابکی LTE و WiMAX، هستند (شکل ۲). در حالت معمولی خواه طول یک صفحه‌چابکی DSRC، کوتاه‌تر از سایرین باشد یا خیر، بسته‌ها می‌باشند به طور کامل طول یک صفحه‌چابکی

پروتکل ارتباطی بی‌سیم DSRC: این استاندارد به عنوان اصلی‌ترین استاندارد برای ارتباط در محیط‌های خودرویی و با نام IEEE802.11p پذیرفته شده است که امکان ارتباط بین خودروهای در حال حرکت و با فاصله ۳۰۰ متر با نرخ ارسال داده ۲۷Mbps فراهم می‌سازد. در محیط‌های باز این فاصله ارتباطی تا ۱۰۰۰ متر افزایش می‌یابد. [۱۴]

۲-۲- کارهای مرتبط

Mario Gerla در [۲] به معروفی یک فریم ورک توزیع شده برای بهره‌وری حداکثری از DSRC در کنار باند کانال تلویزیونی (TVWS)^{۱۷} پرداخته و از طریق این کانال تلویزیونی که برد بسیار زیادی دارد برای پخش بسته‌های داده اورژانسی استفاده می‌کند. در این فریم ورک از DSRC برای روبدول کردن بسته‌های کنترلی و دوباره فرستادن بسته‌های اورژانسی از دست رفته بهره جسته است و نهایتاً توانسته میزان تأخیر را کاهش و به میزان تحویل بسته‌های بالاتری دست یابد. Gerla در محيط همچنین در [۲] به معروفی و امکان‌سنگی استفاده از شبکه LTE در محيط VANET پرداخته و فقط برای دریافت داده از اینترنت خودروها را مجاز به استفاده از شبکه LTE دانسته، درحالی که ارتباطات بین خودرویی و بین خودرو و ایستگاه کنار جاده‌ای همچنان از طریق DSRC انجام می‌گیرد. در این مقاله ارزیابی این روش مورد بررسی قرار نگرفته است. در [۱۵] و [۱۶] نویسنده‌گان به پیش‌گذردگی در شبکه‌های VANET از طریق بهبود عملکرد رابطه‌های ارتباطی در لایه کنترل دسترسی میانی مربوط به تک رادیویی خودرو پرداخته‌اند و با استفاده از کانال‌های چندگانه^{۱۸} توانسته‌اند میزان برخورد و ازدحام را کاهش داده و نهایتاً به گذردگی بالاتری در شبکه دست یابند. در تمامی موارد ذکر شده یا کارهای صورت گرفته بر روی یک رادیویی بوده و یا در صورت استفاده از چند رادیو مدل صفحه‌چابکی جداگانه و به صورت ساده‌ترین شکل یعنی مدل پخش بسته‌ها بین صفحه‌چابکی رادیویی تصادفی بوده است. که در این مقاله مدل صفحه‌چابکی دیگر مورد استفاده قرار گرفته است و نقطه تفاوت اصلی این روش با سایرین می‌باشد. [۱۱] [۹]

۳- روش پیشنهادی و تئوری صفحه‌چابکی

۳-۱- روش پیشنهادی

با توجه به ویژگی‌های اختصاصی که هر کدام از تکنولوژی‌های دسترسی بی‌سیم که در قسمت "پیش‌زمینه" بیان شد دارند و فایده و هزینه هر کدام از آن‌ها، در روش جدید چند رادیویی ارائه شده به دنبال بهره‌وری حداکثری از این تکنولوژی‌ها بر روی هر کدام از خودروها هستیم. سناریویی را در نظر بگیرید که همه خودروها بر روی هر کدام از خودروها هستیم. سناریویی که از تکنولوژی‌های DSRC، WiMAX و LTE پشتیبانی می‌کنند هستند. خودروها می‌توانند برای ارتباطات V2V در بین خودشان یک شبکه نظری به نظری تشکیل داده و بسته‌های داده را با استفاده از رابطه‌های DSRC و یا WiMAX خود جابجا کنند. همچنین اگر خودرویی در طیف رادیویی مناسب برای ارسال بسته نبود، خودروی فرستنده می‌تواند از طریق رابطه ارتباطی LTE خود پس از اتصال به نزدیک‌ترین ایستگاه مخابراتی شروع به ارسال بسته‌های داده نماید و خودروی گیرنده نیز بسته‌های داده را از آن ایستگاه رادیویی دریافت نماید. همچنین خودروها می‌توانند با ایستگاه‌های کار جاده‌ای (RSU) نیز از طریق رابطه DSRC خود ارتباط داشته باشند. همگی این ارتباطات در حالی رخ می‌دهد که خودروها با سرعت در حال حرکت هستند و به دلیل زیاد و کم شدن فاصله آن‌ها نسبت به یکدیگر مدت زمان اتصال کوتاه هست. بنابراین می‌بایست در همین مدت زمان کوتاه به گونه‌ای بسته‌ها را روبدول کنیم که میانگین

$$W_{k,i} = \frac{k+1}{\mu} \quad (1)$$

در ادامه نشان می‌دهیم که چگونه می‌توان مقادیر شرطی مورد انتظار و توابع مولد آن‌ها را محاسبه کرد. [۱]

اگر فرض کنیم که $W_{-1,i} = 0$ باشد به معادله زیر می‌رسیم. (فرض می‌کنیم $\lambda + 2\mu = 1$)

$$W_{k,i} = 1 + \lambda W_{k,i+1} + \mu W_{k,i-1} + \mu W_{k-1,i+1} \quad (2)$$

$$k \geq 0, i \geq 1$$

همچنین اگر فرض کنیم که برای $k \leq N - 2$ معادله زیر برقرار باشد

$$W_{k-N+1,2N-3} = (k+1)/\mu \quad (3)$$

شرط مزدی به شکل زیر تعریف می‌گردد: [۱]

$$W_{k,0} = 1 + \lambda W_{k,1} + \mu W_{k-1,1} + \mu W_{k-N+1,2N-3}, k \geq 0 \quad (4)$$

$$W_{k,i} = \frac{k+1}{\mu}, 0 \leq k \leq N-2, i \geq 0 \quad (5)$$

و نهایتاً

$$\lim_{i \rightarrow \infty} W_{k,i} \leq \frac{k+1}{\mu}, k \geq 0 \quad (6)$$

در مرحله بعد برای حل معادلات (۳) تا (۶) می‌بایست تابع مولد جواب را پیدا کنیم. در ابتدا برای $i \geq 0$ و برای یک مقدار پیچیده t با $|t| < 1$ تابع $G_i(t) = \sum_{k=0}^{\infty} W_{k,i} t^k$ را در نظر می‌گیریم. دوم اینکه، $i \geq 1$ را برای یک مقدار ثابت λ در نظر می‌گیریم. سپس با جایگذاری در معادله (۳) و به یاد داشتن اینکه $W_{-1,i} = 0$ و پس از ساده‌سازی به معادله زیر می‌رسیم. [۱] [۱۵]

$$(\lambda + t\mu)G_{i+1}(t) - G_i(t) + \mu G_{i-1}(t) = \frac{-1}{1-t}, i \geq 1 \quad (7)$$

با نگاهی به مقدار داده شده برای t جواب قسمت ناهمگن معادله (۷) به صورت زیر است:

$$G_i(t) = \frac{1}{\mu(1-t)^i}, i \geq 0 \quad (8)$$

با نگاهی به قسمت همگن می‌بینیم که هر دو جواب به صورت زیر مشخص می‌شوند: [۱]

$$f_-(t) = \frac{1-\sqrt{1-4\mu(\lambda+\mu t)}}{2(\lambda+\mu t)} \quad (9)$$

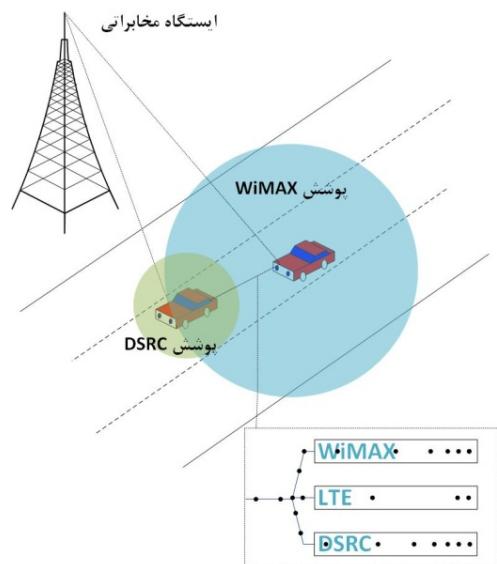
و

$$f_+(t) = \frac{1+\sqrt{1-4\mu(\lambda+\mu t)}}{2(\lambda+\mu t)} \quad (10)$$

و بدین ترتیب مسئله حل می‌شود. ضرایب معادلات همگن و ناهمگن نیز به همین طریق به دست می‌آید.

طی کنند تا ارسال شوند، در حالی که در روش پیشنهادی چند رادیویی بسته‌هایی که در صفحه بلندتر هستند اجازه دارند به صفحه که کوتاه‌تر از سایرین هست پیووندند به این امید که زودتر ارسال شوند. البته ممکن است با توجه به ویژگی‌های متفاوت رابطه‌های ارتباطی، بسته‌ای که صفحه خود را عوض کرده مجدداً صفحه قلی خود را کوتاه‌تر ببیند و به همان قلی برگرد و یا حتی به صفحه سوم پیووند. در هر صورت بسته اجازه دارد تا زمان ارسال عمل چابکی (رفتن به صفحه کوتاه‌تر)، انجام دهد و فرض مسئله این است که این عمل هزینه‌بر نیست. [۴]

برای ساده‌تر شدن محاسبات و مدل کردن مسئله، تعداد صفحه را ۲ در نظر می‌گیریم. حال آنکه راهکار حل مسئله برای هر تعداد دلخواه از صفحه معتبر است. توجه به این مسئله حائز اهمیت است که میزان اختلاف دو صفحه عنوان مقدار آستانه^{۱۷} که بسته‌ها اجازه انجام عمل چابک پیدا می‌کنند بسیار حائز اهمیت است. به عنوان مثال سناریویی را در نظر بگیرید که میزان آستانه را ۱ در نظر بگیریم، بنابراین زمانی که دو صفحه دارای طول برابر شوند بسته جدید و روودی مرتباً در انتهای دو صفحه جایجا می‌شود و ناپایدار است.



شکل ۲- ورود بسته‌ها به سه صفحه جهت ارسال

هدف ما به دست آوردن میزان زمان انتظار^{۱۸} یک بسته در صفحه به ازای هر مقدار دلخواه آستانه N است. بنابراین اصل زیرمبنا تمامی مفروضات و تحلیل‌های ماست: [۱]

"در هر لحظه، زمان انتظار یک بسته به تعداد کل بسته‌های موجود در سیستم (تمامی صفحه) و به اینکه چه تعداد بسته جلوی او در صفحه قرار دارند وابسته است. با این حال، در صورت داشتن این دو مقدار، زمان انتظار به چگونگی تقسیم بسته‌ها در دو صفحه ندارد" منطبق با اصل فوق برای $0 \leq K \leq N$ می‌توانیم $W_{k,i}$ را زمان انتظار یک بسته در نظر می‌گیریم که K بسته در جلویش در صفحه خودش قرار دارند و از تعداد کل بسته‌های پشت سرش در همان صفحه است. همچنین در هر دو صفحه تعدادی بسته وجود دارد که در یک صفحه تا موقعیت ماکزیمم $\{k - N + 3, 0\}$ و بیشتر از این مقدار در صفحه دیگر اشغال است. نرخ ورود بسته‌ها به سیستم λ و نرخ سرویس دادن به بسته‌ها μ می‌باشد. توجه داشته باشید که براساس مفروضات مسئله اگر یک بسته تعداد K بسته را در صفحه کنونی اش جلوی خودش ببیند، برای بعضی مقادیر K که در آن $2 \leq N - 2 \leq K \leq N + 2$ باشد، تمامی موقعیت‌ها در صفحه دیگر از موقعیت ۱ و بیشتر تا موقعیت K-1 اشغال است. بنابراین واضح است که: [۱]

$$i \geq K \leq N - 2 \leq 0 \leq i \leq N - 2$$

۳-۳- توزیع ایستا

۴- شبیه‌سازی و ارزیابی روش

شبیه‌سازی تقليیدی از عملکرد فرایند یا سیستم واقعی با گذشت زمان است. از آنجا که سیستم مسئله مورد نظر یک سیستم گسسته است بنابراین با شبیه‌سازی گسسته پیش‌آمد^{۲۵} سروکار داریم. در این بخش به بیان روش شبیه‌سازی و نحوه اجرای آن، نتایج حاصل از اجرای شبیه‌سازی و نهایتاً ارزیابی نتایج می‌پردازیم.

۴-۱- روش شبیه‌سازی

با توجه به پارامترهای مسئله مورد بررسی، برای شبیه‌سازی سیستم، جدول ۱ که نمونه‌ای از جدول شبیه‌سازی بسته‌های ارسالی در شبکه با مفروضات بیان شده است را تشکیل داده‌ایم.

جدول ۱- نمونه‌ای از جدول شبیه‌سازی بسته‌های داده در شبکه

شماره بسته	زمان ورود	صف ورودی بسته	زمان شروع خدمت	انجام Jockey	مدت خدمت‌دهی	زمان پایان خدمت
۱	۰	۱	۰	۰	۶	۶
۲	۱	۲	۱	۰	۴	۴
۳	۲	۱	۶	۰	۴	۱۰
۴	۵	۱	۱۰	۱	۳	۱۳
۴	۵	۲	۵	۰	۵	۱۰
۵	۷	۲	۸	۰	۵	۱۳
۶	۹	۱	۱۰	۰	۷	۱۷

برای اجرای شبیه‌سازی از نرم‌افزار متلب [۱۲] با تنظیمات بیان شده در جدول ۲ استفاده شده است:

جدول ۲- جدول تنظیمات شبیه‌سازی بسته‌های شبکه

مقدار تنظیم شده	نوع پارامتر تنظیم
۶۰	تعداد تکرار شبیه‌سازی
۵۰.۰۰۰	تعداد بسته در هر اجرا
Poisson	توزیع بسته‌های ورودی
Exponential	توزیع خدمت‌دهی به بسته‌ها
۱	نرخ خدمت‌دهی μ
۰.۵	نرخ ورود بسته‌ها λ
۳	مقدار حد آستانه N
۱۱	تعداد دفعات افزایش μ
(۰.۰ و ۰.۵)	احتمال اختصاص اولیه به صفت کوتاه‌تر P
۰ (۰.۰۵ و ۰.۱) اولین- ورودی اولین- خروجی(FIFO)	نظم ورود و خروج از صفحه

بسته‌های داده با توزیع پواسون تولید شده و مدت زمان سرویس گرفتن (ارسال شدن) از توزیع نمایی پیروی می‌کند. شبیه‌سازی را با مقدار ثابت بسته‌های داده از ۱ تا ۵۰.۰۰۰ و تا ارسال آخرین بسته در هر دفعه اجرا انجام داده‌ایم، همچنین نرخ ورود بسته‌ها و نرخ خدمت‌دهی (سرویس گرفتن بسته‌ها) را به تعداد ۱۱ دفعه در شبیه‌سازی‌های متفاوت افزایش داده‌ایم تا نتایج مختلف حاصل گردد و کارایی روش را بتوان ارزیابی کرد. برای بدست آوردن تأخیر در هر کدام از مقدار پارامترهای λ و μ شبیه‌سازی را ۵۰ بار تکرار کرده‌ایم. مدت زمان سرویس گرفتن را در محاسبه زمان تأخیر لحاظ نکرده‌ایم. همچنین برای بسته‌های داده اولویت در نظر گرفته نشده و همه بسته‌ها یکسان می‌باشند. میزان آستانه اختلاف صفحه

در این قسمت نشان می‌دهیم که چگونه می‌توان رفتار بسته‌ها را در حالت توزیع ایستا (پایدار)^{۱۹} برای مدل دو صف موازی بدون حافظه^{۲۰}، زمانی که عملیات چابکی با گذر از حد آستانه (اختلاف بین دو صف) از مقدار N که $N \leq 3$ است اتفاق می‌افتد را محاسبه کرد. برای سادگی محاسبات پارامتر N را در این بخش حذف می‌کنیم.

ما از محاسبات براساس Matrix-geometric برای بدست آوردن توزیع پایدار استفاده خواهیم کرد. این تکنیک به تجزیه فضای حالت ۲۱ نیاز دارد. ضمناً این تجزیه یکتا نیست و روش تجزیه‌ای که در این مقاله استفاده شده است مناسب‌ترین حالت برای محاسبات ماست.

از λ به عنوان نرخ ورودی و μ به عنوان نیازمندی سرویس مورد انتظار استفاده می‌کنیم. همچنین قیاس λ و μ را به صورت $\lambda + 2\mu$ و $\lambda + \mu$ برای اکنون π_i را احتمال پایداری در صورتی که i مشتری در یکی از صفحه‌ها و j مشتری در صف دیگر است در نظر می‌گیریم. این تعداد شامل مشتریان در حال خدمت گرفتن بدون از دست دادن "عمومیت"^{۲۲} می‌شود.

فرض کنید که $j \leq i$ برای $0 \leq i \leq N$ باشد و (i) به عنوان مجموعه‌ای از حالات (j,i) برای $j \leq i + N - 1$ در نظر گرفته شود. این N حالت را می‌توانیم به صورت (i,j), (i,i+1), ..., (i,N-1) نویسیم. همچنین π_i را به عنوان بردار خطی^{۲۳} از احتمالات پایدار حالتی که در بالا بصورت L(i) بیان شد در نظر می‌گیریم. برای سادگی در محاسبات و یافتن ماتریکس‌های Q_0 و Q_1 و Q_2 در $R^{N \times N}$ برای $1 \leq i \leq N$ معادله زیر را بدست می‌آوریم.

$$\pi_i Q_0 + \pi_{i+1} Q_1 + \pi_{i+2} Q_2 = 0 \quad (11)$$

بنابراین مقادیر $\lambda_1 = \lambda(1 + p)/2$ و $\lambda_2 = \lambda(1 - p)/2$ را در نظر می‌گیریم. و ماتریس به شکل زیر می‌توانیم تعریف نماییم.

$$Q_0(i,j) = \begin{cases} \lambda_1 & i = 2, \dots, N-1, j = i-1 \\ \lambda & i = N, j = N-1 \\ 0 & \text{سایر حالات دیگر} \end{cases} \quad (12)$$

ماتریس فوق را برای سادگی به شکل ماتریس زیر نیز می‌توانیم بنویسیم.

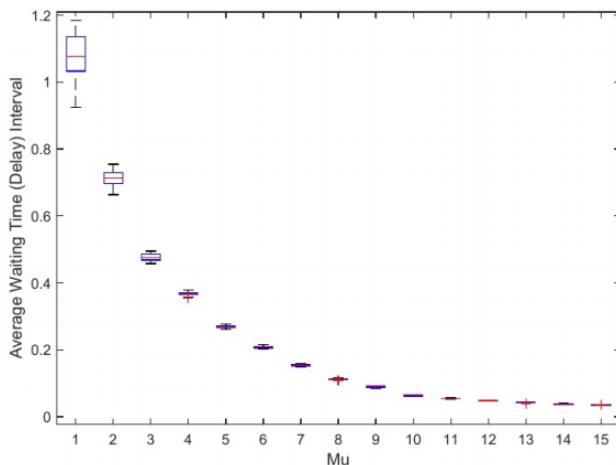
$$Q_0 = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ \lambda_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \lambda \end{pmatrix} \quad (13)$$

به همین ترتیب مقادیر Q_1 و Q_2 نیز محاسبه می‌گردد. قبل از [۱۶] نشان داده شده است که وقتی توزیع پایدار وجود داشته باشد، تمامی مقادیر ویژه‌ی R بصورت یکتا می‌باشد.علاوه، از آنجا که یک ردیف از Q_0 تماماً صفر است، متعاقباً با حالت یکسانی برای ماتریس R حاصل یکی از مقادیر ویژه آن که می‌توانیم w_1 بنامیم همگی صفر هستند. بنابراین طیف شعاعی R^{24} برای λ^2 می‌باشد. [۱۵]

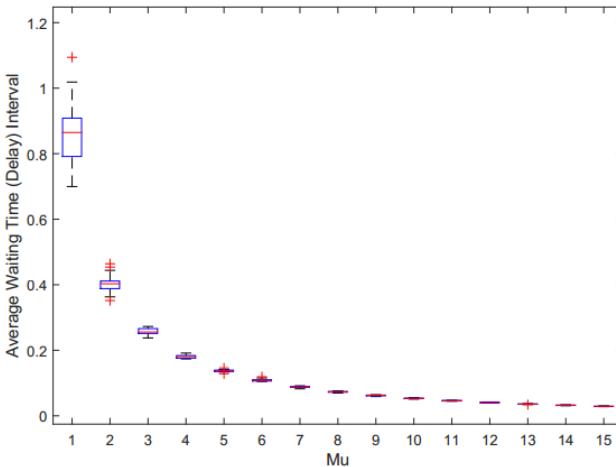
[۱۶]

به صف کوتاه‌تر می‌پیوندد. این کار تا زمانی ادامه می‌یابد که بسته‌ها به طور کامل ارسال شوند.

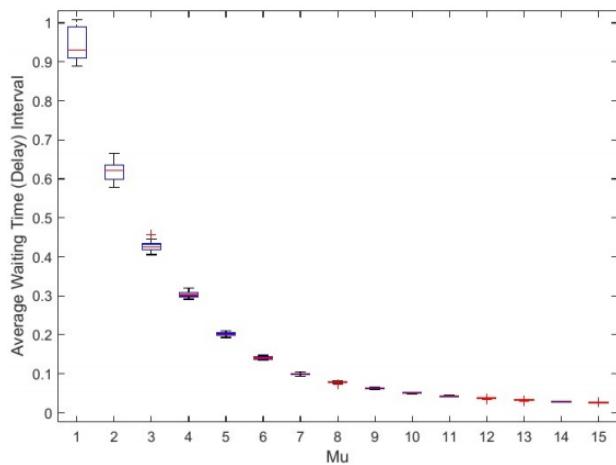
برای انجام عمل جابجایی بسته‌ها و رفتن بسته از صف بلندتر به صف کوتاه‌تر برابر با ۳ در نظر گرفته شده است ($N = 3$).



شکل ۵- میانگین زمان تأخیر بدون مدل صف چابکی و با مقدار λ ثابت و μ متغیر و $p=0.5$ و $N=3$



شکل ۶- میانگین زمان تأخیر با استفاده از مدل صف چابکی و با مقدار λ ثابت و μ متغیر و $N=3$ و $p=0.5$

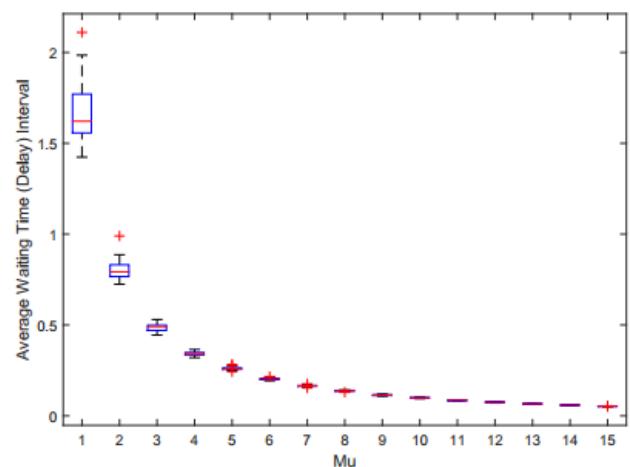


شکل ۷- میانگین زمان تأخیر بدون مدل صف چابکی و با مقدار λ ثابت و μ متغیر و $N=3$ و $p=1$

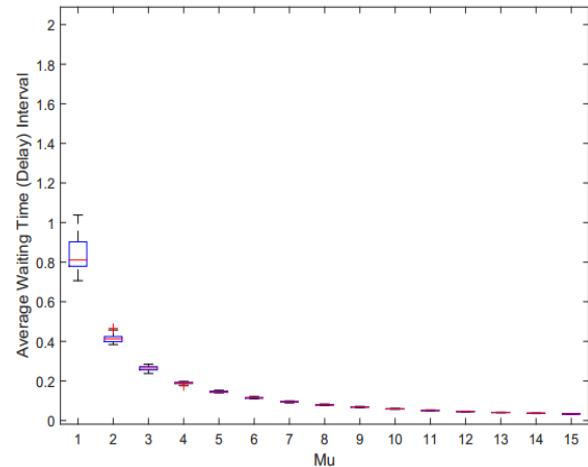
۲-۴- نتایج شبیه‌سازی

در شکل‌های این قسمت نتایج مربوط به شبیه‌سازی‌ها در حالات مختلف ارایه گردیده است. این حالات شامل ثابت نگه داشتن پارامتر λ و تغییر در پارامتر μ به ازای سه مقدار مختلف پارامتر P (مقادیر ۰ و ۰.۵ و ۱) در دو حالت "چابکی" و "غیر-چابکی" است.

در شکل ۳ نمودار بازه‌ی میانگین زمان انتظار برای یک سیستم صف دوتایی، بدون در نظر گرفتن "چابکی" و با مقدار ثابت $\lambda=1$ و مقدار متغیر μ ، بدین صورت که مقدار اولیه برابر $\mu_1 = 0.5$ (برای صف اول) و $\mu_2 = 0.6$ (برای صف دوم)، همچنین مقدار حد آستانه $P=0$ و $N=3$ اجرا شده است.



شکل ۳- میانگین زمان تأخیر بدون مدل صف چابکی و با مقدار λ ثابت و μ متغیر و $N=3$ و $p=0$



شکل ۴- میانگین زمان تأخیر با استفاده از مدل صف چابکی و با مقدار λ ثابت و μ متغیر و $N=3$ و $p=0$

در شکل ۴، همین آزمایش مجدد تکرار و این بار حالت "چابکی" در نظر گرفته شده است. با این حساب، بسته‌های در انتظار ارسال در هر کدام از صفحه‌ها و در هر زمانی (و نه فقط در تخصیص‌های اولیه بسته‌ها به صفحه‌ها)، اگر تفاوت تعدادشان در دو صفحه مساوی و یا بیشتر از ۳ شود، آخرین بسته داده از صفحه بلندتر

شایان ذکر است این تعادل بار بر روی حالت چاپکی تغییری اعمال نکرده است. همچنین در تعداد بسته‌های کم، مدل چند رادیویی با مدل تک رادیویی تفاوت چندانی ندارد. چراکه بسته‌ها از طریق یک رادیو و یک صفحه ارسال هم بدون تأخیر و یا با تأخیر بسیار ناچیز ارسال می‌گردند، اما زمانی که تعداد بسته‌های درون سیستم افزایش می‌یابد مدل چند رادیویی بهتر از مدل تعادلی عمل می‌کند. شایان ذکر است برای انجام شبیه‌سازی از آنچاکه سایر روش‌های قبلی ارسال داده در شبکه‌های VANET از رابطه‌های ارتباطی تکی (یک رادیو) استفاده شده بود، و ماهیت روش ارائه شده در این مقاله چند رادیویی و استفاده از چند رابط ارتباطی بی‌سیم است بنابراین برای مقایسه روش‌های روشی را در نظر نگرفتیم، چراکه بالا بودن گذردهی در شبکه‌ای که خودروها مجهز به چند رادیو هستند نسبت به زمانی که از یک رادیو استفاده می‌کنند امری بدینه است. بنابراین مبنای مقایسه را بر مدل‌هایی از شبکه‌های VANET که از بیش از یک رادیو استفاده می‌کرده‌اند قرار داده‌ایم. قبل از قسمت مقدمه راجع به این روش‌ها صحبت شده است.

۵- نتیجه‌گیری

در این مقاله به مدل سازی تأخیر در ارسال بسته‌هایی که بین خودروها در شبکه‌های VANET رد و بدل می‌شوند، به کمک فن آوری‌های چندگانه بی‌سیم پرداختیم و توانستیم با استفاده از مدل صفحه چاپکی تأخیر نهایی را کاهش دهیم. مدل صفحه چاپکی به بسته‌ها این اجازه را می‌دهد تا از یک صفحه بلندتر به صفحه کوتاه‌تر تغییر مکان دهند، بنابراین مدت زمان انتظار بسته‌ها در هر صفحه ارسالی کاهش و با کاهش تأخیر ارسال بسته‌ها موجب افزایش گذردهی شبکه می‌گردد. در این روش از چند رادیو با تکنولوژی‌های دسترسی بی‌سیم مقاومت شامل DSRC، LTE، WiMAX می‌کنیم و خودرو می‌تواند بسته‌های داده ارسالی را بر روی هر کدام از این رابطه‌های ارتباط بی‌سیم قرار داده و ارسال کند. همچنین در این مقاله به بررسی حالات پرداختیم که بسته‌ها از ابتدای توزیع با احتمالات مختلف ($P=0$, $P=0.5$, $P=1$) وارد صفحه کوتاه‌تر شوند و نتیجه تخصیص‌های گوناگون را در هر دو روش استفاده از مدل صفحه چاپکی و عدم استفاده از این مدل بر روی بازه‌های میانگین زمان انتظار بسته‌ها مشاهده کردیم. علاوه بر این، پس از فرموله کردن مسئله به محاسبه توزیع حالت ایستا پرداختیم و رفتار بسته‌ها را در دراز مدت بررسی کردیم. سپس به بررسی نتایج حاصل از شبیه‌سازی پرداخته شد. نتایج شبیه‌سازی حاکی از کارایی چشمگیر این روش در مقایسه با روش‌های کارهای مشابه قبلی که عموماً بسته‌های داده را به طور تصادفی در بین صفحه‌های ارسال پخش می‌کردند، است. این روش بخصوص زمانی که خودروها بسته‌های داده بسیاری جهت ارسال دارند، که منجر به شلوغ تر شدن و طولانی تر شدن صفحه‌های ارسال داده در بین خودروها می‌گردد و با در نظر گرفتن $N=3$ کارایی چشمگیرتری از خود نشان داد.

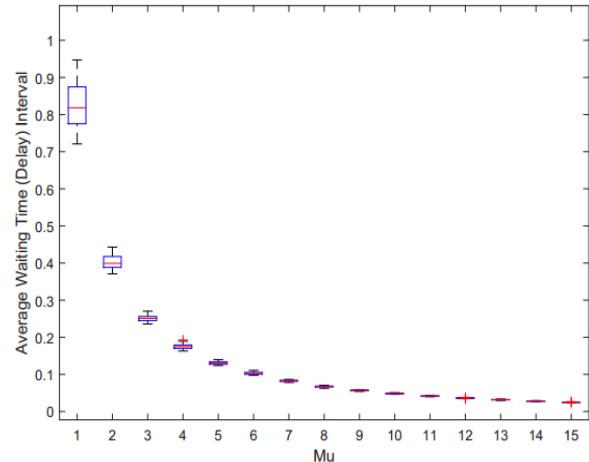
مراجع

[1] R. Hassin, and M. Haviv. To Queue or Not to Queue: Equilibrium Behavior in Queueing Systems. NY, USA: Springer US, 2003.

[2] M. Gerla, and et. al., "Content Distribution in VANETs," *Vehicular Communications*, vol. 1, no. 1, pp. 3-12, 2014.

ضمناً ممکن است بسته‌ای که یکبار عمل "چاپکی" را انجام داده، دوباره با گذشت زمان مجدد طول دو صفحه از حد آستانه عبور کند و به عنوان آخرین بسته مجدداً به صفحه قبلی خود برگردید. بنابراین در دفعات انجام عمل "چاپکی" برای هر بسته محدودیتی لحاظ نگرددیده است.

در ادامه در شکل ۵ الی شکل ۸ مقایسه مدل صفحه چاپکی با غیرچاپکی، با ثابت در نظر گرفتن λ و متغیر بودن μ (به همان ترتیبی که در مورد شکل ۳ بیان شد) و به ازای دو مقدار متفاوت $P=0.5$ و $P=1$ ارایه شده است. در شکل ۷ و شکل ۸ نمودارهای فوق با لحاظ کرده $P=1$ ارایه شده است. همانطور که مشخص است، در این حالت نیز نمودار با حالت "چاپکی" نسبت به "غیرچاپکی" از بازه میانگین زمان انتظار کمتری برخوردار است.



شکل ۸- میانگین زمان تأخیر با استفاده از مدل صفحه چاپکی و با مقدار λ ثابت و $N=3$ و $p=1$ و $p=0.5$ متفاوت

۴-۳- ارزیابی نتایج شبیه‌سازی

از مقایسه شکل ۳ و شکل ۴ مشخص است که در حالت لحاظ کردن "چاپکی" بازه میانگین زمان انتظار بسته‌ها، به مقدار قابل توجهی کاهش یافته است. در نهایت در هر دو حالت، با افزایش نرخ خدمتدهی، یعنی پردازش سریع تر و ارسال سریع تر بسته‌ها، با ثابت ماندن نرخ تولید بسته، مقدار بازه میانگین زمان انتظار کاهش و به صفر نزدیک می‌گردد، اما در هر مقداری از λ حتی در مقداری بالای آن، عملکرد راهکار "چاپکی" بهتر از "غیرچاپکی" بوده و با کاهش بازه میانگین زمان انتظار همراه است.

در مدل صفحه چاپکی بسته‌ها اجازه دارند در صورت مشغول بودن یک رابطه ارتباطی بی‌سیم از صفحه خود خارج شوند و به صفحه کوتاه‌تری از بسته‌ها که از طریق رابطه ارتباطی بی‌سیم دیگر خودرو فرستاده می‌شوند بیرونند. همان‌طور که از شکل‌های ۳ و ۴ مشخص است، هنگامی که بسته‌ها کاملاً تصادفی بین دو صفحه تقسیم شوند ($p=0$) تقاضاً (بهبود عملکرد) استفاده از مدل صفحه چاپکی نسبت به عدم استفاده از این روش، در میانگین مدت زمان انتظار بسته‌ها بسیار بیشتر از $P=1$ حالتی است که در شکل ۷ و ۸ دیده می‌شود. در واقع میزان تأخیر در حالت $P=1$ در هر دو روش در شکل ۷ و ۸ کمتر دستخوش تغییر است.

دلیل این امر، این است که وقتی تمامی بسته‌ها در حالت غیرچاپکی از ابتدای تقسیم بین دو صفحه، به صفحه کوتاه‌تر اختصاص یافته‌اند به نوعی موجب تعادل بار در توزیع بسته‌ها شده است که به شدت میانگین زمان تأخیر را کاهش می‌دهد. هرچند در همین حالت هم باز استفاده از مدل صفحه چاپکی عملکرد بهتری را در پی داشته است.

- [16] R. Hassin, and M. Haviv, "Equilibrium Strategies and the Value of Information in a Two Line Queueing System with Threshold Jockeying," *Stochastic Models*, vol. 10, no. 2, pp. 415-435. 1994.
- [17] P. Manzoni, et. al., "Mobility Models for Vehicular Communications," in *Vehicular ad hoc Networks*, 1st ed. Springer International Publishing, 2015, ch. 11, pp. 309-333.
- [18] S. Gonzalez, and V. Ramos, "Preset Delay Broadcast: A Protocol for Fast Information Dissemination in Vehicular Ad Hoc Networks (VANETs)," *EURASIP Journal on Wireless Communications and Networking*, no. 1, pp.1-13, 2016.
- [19] A. M. Said, M. Marot, A. W. Ibrahim, and H. Afifi, "Modeling Interactive Real-time Applications in VANETs with Performance Evaluation," *Computer Networks*, no. 104, pp. 66-78, 2016.
- محمد امین کشتکار فارغ‌التحصیل کارشناسی ارشد در رشته مهندسی فناوری اطلاعات از دانشگاه تهران است. ایشان سابقه چند سال فعالیت و پژوهش در زمینه‌های امنیت کامپیوتری، رایانش ابری، سیستم‌های اطلاعاتی و تجارت الکترونیک در پژوهشکده‌ها، شرکت‌ها و مراکز مختلف دانشگاهی را در کارنامه خود دارد و مقالات متعددی در این حوزه‌ها در کنفرانس‌ها و مجلات علمی منتشر کرده‌اند. آدرس پست‌الکترونیکی ایشان عبارت است از:**
- a.keshtkar@ut.ac.ir
- احمد خونساری مدرک کارشناسی مهندسی برق و کامپیوتر خود را از دانشگاه شهید بهشتی دریافت کرد و پس از آن دوره کارشناسی ارشد را در دانشگاه علم و صنعت به اتمام رساند و همچنین مدرک دکترا در علوم کامپیوتر را از دانشگاه گلاسگوی انگلستان دریافت کرد. ایشان در حال حاضر دانشیار دانشکده برق و کامپیوتر دانشگاه تهران و پژوهشگر در پژوهشگاه دانش‌های بنیادی است. علایق پژوهشی ایشان ارزیابی و مدل‌سازی کارائی، شبکه‌های مخابراتی و سیستم‌های توزیع شده و معماری کامپیوتر است. آدرس پست‌الکترونیکی ایشان عبارت است از:**
- ak@ipm.ir
- ابوالفضل دیانت کارشناسی خود را از دانشکده مهندسی برق و کامپیوتر دانشگاه تهران اخذ کرد و پس از آن مدرک کارشناسی ارشد خود را در گرایش مخابرات از دانشگاه صنعتی شریف گرفت. ایشان اکنون در حال گذراندن دوره دکترای تخصصی خود در دانشگاه تهران می‌باشد. اصلی‌ترین علایق پژوهشی ایشان سیستم‌های مخابراتی، حریم خصوصی آماری در شبکه و مدل‌سازی تحلیلی شبکه‌های بی‌سیم می‌باشد. آدرس پست‌الکترونیکی ایشان عبارت است از:**
- a.diyanat@ut.ac.ir
- [3] N. Naito, and et. al., "Interplay between TVWS and DSRC: Optimal Strategy for QoS of Safety Message Dissemination in VANET," Proc. *Computing, Networking and Communications (ICNC)*, 2013.
- [4] Z. H. Mir, and F. Filali, "LTE and IEEE 802.11p for Vehicular Networking: A Performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 89, 2014.
- [5] N. Lu, and et. al., "A Dedicated Multi-Channel MAC Protocol Design for VANET with Adaptive Broadcasting," *Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 2010.
- [6] Y. Zang, and et. al., "A Novel MAC Protocol for Throughput Sensitive Applications in Vehicular Environments," Proc. IEEE 65th Vehicular Technology Conference, pp. 2580-2584, 2007.
- [7] T. Wang, L. Song, and Z. Han, "Collaborative Data Dissemination in Cognitive VANETs with Sensing-throughput Tradeoff," *Communications in China (ICCC)*, pp. 41-45, 2012.
- [8] N. J. Kirsch, and B. M. O'Connor, "Improving the Performance of Vehicular Networks in High Traffic Density Conditions with Cognitive Radios," *Intelligent Vehicles Symposium (IV)*, pp. 552-556, 2011.
- [9] A. J. Ghandour, K. Fawaz, and H. Artail, "Data Delivery Guarantees in Congested Vehicular Ad Hoc Networks Using Cognitive Networks," *Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 871-876, 2011.
- [10] M. Chaqfeh, A. Lakas, and I. Jawhar, "A Survey on Data Dissemination in Vehicular Ad Hoc Networks," *Vehicular Communications*, pp. 214-225, 2014.
- [11] K. D. Singh, P. Rawat, and J. Bonnin. "Cognitive Radio for Vehicular Ad Hoc Networks (CR-VANETs): Approaches and Challenges," *EURASIP Journal on Wireless Communications and Networking*, pp. 1-22, 2014.
- [12] F. J. Ros, J. A. Martinez, and P. M. Ruiz. "A Survey on Modeling and Simulation of Vehicular Networks: Communications, Mobility, and Tools," *Computer Communications*, vol. 43, pp. 1-15, 2014.
- [13] J. Santa, A. F. Gómez-Skarmeta, and M. Sánchez-Artigas, "Architecture and Evaluation of a Unified V2V and V2I Communication System Based on Cellular Networks," *Computer Communications*, vol. 31, no. 12, pp. 2850-2861, 2008.
- [14] J. B. Kenney, "Dedicated Short-range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162-1182, 2011.
- [15] Y. Zhao, and W. K. Grassmann, "Queueing Analysis of a Jockeying Model," *Operations research*, vol. 43, no. 3, pp. 520-529, 1995.

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۵/۰۴/۱۶

تاریخ اصلاح: ۱۳۹۵/۰۶/۱۳

تاریخ قبول شدن: ۱۳۹۵/۰۷/۰۱

نویسنده مرتبط: محمد امین کشتکار، دانشکده برق و کامپیوتر، دانشگاه
تهران، تهران، ایران.

^۱Vehicular Ad Hoc Networks^۲Mobile Ad Hoc Networks^۳Roadside Unit^۴IEEE^۵Vehicle-To-Vehicle^۶Vehicle-To-Infrastructure^۷Wireless Access in Vehicular Environment^۸Dedicated Short-Range Communication^۹Long-Term Evolution^{۱۰}Jockey^{۱۱}Broadband Cloud Communication^{۱۲}TV White Space Channel^{۱۳}Multi-Channel MAC^{۱۴}Balking^{۱۵}Reneging^{۱۶}Jockeying^{۱۷}Threshold^{۱۸}Waiting Time^{۱۹}Stationary State^{۲۰}Memoryless Parallel Queue Model^{۲۱}State Space^{۲۲}Generality^{۲۳}Row-Vector^{۲۴}Spectral Radius^{۲۵}Discrete-Event Simulation^{۲۶}First-In First-Out

شناسائی عملی کلیک‌های هرز در وب با استفاده از الگوریتم‌های دسته‌بندی

سجاد ظریف‌زاده مهرده فلاح

دانشکده برق و کامپیوتر، دانشگاه یزد، یزد، ایران

چکیده

امروزه اکثر سرویس‌های اینترنتی از بازخورد کاربران برای بهبود کیفیت سرویس‌دهی به آنان استفاده می‌نمایند. به عنوان مثال، موتورهای جستجو از اطلاعات کلیک کاربران به عنوان یک فاکتور مهم در فرآیند رتبه‌بندی نتایج جستجو بهره می‌برند. از همین‌رو، برخی وب سایت‌ها برای کسب رتبه بالاتر در بین مجموعه نتایج جستجو به انجام کلیک بر روی نتایج خود می‌پردازن. چون این کلیک‌ها توسط کاربران واقعی انجام نگرفته، اصطلاحاً به آنها کلیک‌های هرز گفته می‌شود. برای این منظور، وب سایت‌ها معمولاً از برنامه‌های نرم‌افزاری به نام "ربات‌ها" استفاده می‌کنند تا به صورت خودکار و توزیع شده به انجام این کار پردازن. در این مقاله، روش جدیدی مبتنی بر دسته‌بندی نتایج کاربران جهت شناسائی کلیک‌های هرز به صورت سریع و کارآمد پیشنهاد می‌شود. ما در ابتدا نتایج کاربران را به صورت مجموعه‌ای از ویژگی‌ها مدل می‌کنیم و سپس با اعمال الگوریتم دسته‌بندی پیشنهادی، اقدام به شناسائی نتایج کلیک‌های هرز می‌نماییم. روش مطرح شده با لایق واقعی یک موتور جستجوی فارسی موردن تحلیل قرار گرفته است. نتایج بررسی‌ها نشان می‌دهد که روش پیشنهادی می‌تواند کلیک‌های هرز را با دقیقی بیش از ۹۶٪ تشخیص دهد که در مقایسه با کارهای قبلی در حدود ۵ درصد بهبود از خود نشان می‌دهد.

کلمات کلیدی: کلیک هرز، شناسائی ربات‌ها، ناهنجاری، یادگیری ماشین.

۱- مقدمه

خودکار به ارسال پرس‌وجو و یا انجام کلیک روی لینک‌ها می‌پردازند) به صورت توزیع شده، به انجام حملات مختلف دست می‌زنند. چون این کلیک‌ها توسط کاربران واقعی انجام نمی‌شود، اصطلاحاً به آنها "کلیک‌های هرز" گفته می‌شود. مسئله شناسائی و تفکیک ترافیک تولید شده توسط ربات‌ها از ترافیک کاربران واقعی و نرمال برای موتورهای جستجو بسیار حائز اهمیت است، زیرا وجود ترافیک‌های غیر نرمال علاوه بر تغییر در رتبه‌بندی نتایج جستجو می‌تواند با مصرف پهنه‌ای باند موتور جستجو، افزایش زمان پاسخگویی به کاربران واقعی و تاثیر منفی روی تضمیم‌گیری‌هایی که براساس سابقه و بازخورد کاربران گرفته می‌شود [۱۵]، به موتور جستجو صدمه بزنند.

از سوی دیگر، درآمد اصلی سرویس‌های رایگانی نظریه موتورهای جستجو از سیستم تبلیغات آنلاین آنها می‌باشد. افزایش روز افرون کاربران اینترنتی نیز به رونق این کسب و کار کمک شایانی نموده به نحوی که درآمد حاصل از تبلیغات آنلاین در سال ۲۰۱۵ به ۵۹/۶ میلیارد دلار رسیده است که این مقدار نیست به سال قبل خود، بیشتر از ۱۵٪ رشد داشته است [۶]. از این منظر، موتورهای جستجو به عنوان شبکه‌های تبلیغاتی به نمایش لینک‌های تبلیغاتی اقدام می‌کنند.

امروزه موتورهای جستجو امکان دسترسی سریع، آسان و رایگان را به منابع عظیم اطلاعاتی موجود در سطح اینترنت برای کاربران فراهم می‌آورند. هنگامی که کاربر پرس‌وجوی خود را در موتور جستجو وارد می‌کند، آنها اسناد مرتبط با پرس‌وجوی کاربر را یافته و بر اساس فاکتورهای متعددی نظریه ویژگی‌های متنی [۱۶] و ساختار پیوندی بین صفحات [۲] رتبه‌بندی کرده و به کاربر نمایش می‌دهند. در دهه اخیر، موتورهای جستجو برای بهبود نتایج ارائه شده به کاربران، از کلیک‌های انجام شده روی مجموعه نتایج نیز به عنوان بازخورد مناسبی از سوی کاربران استفاده نموده و آن را در فرآیند رتبه‌بندی اسناد وارد می‌سازند. اکثر کاربران تنها به نتایج نخست (با رتبه بالاتر) توجه می‌نمایند [۳]. این امر می‌تواند موجب سوءاستفاده از موتورهای جستجو و دست‌کاری صفحه نتایج به منظور بالا بردن رتبه برخی صفحات خاص و یا احیاناً خرابکاری شود [۴]. حمله‌کنندگان با استخدام مجموعه‌ای از افراد و یا با استفاده از ربات‌ها (برنامه‌های نرم‌افزاری که به صورت

شناسائی کلیک‌های هرز می‌باشد را معرفی و تشریح می‌کنیم. در بخش چهارم به ارزیابی روش پیشنهاد شده و مقایسه آن با کارهای قبلی می‌پردازیم و نهایتاً در بخش پنجم به جمع‌بندی و ارائه پیشنهاداتی برای ادامه کار خواهیم پرداخت.

۲- کارهای مرتبط

در یک دهه گذشته، شناسائی کلیک‌های هرز در شبکه‌های تبلیغاتی بسیار مورد توجه قرار گرفته است. استون-گراس و همکارانش [۱۲] با استفاده از لگ جستجوی یک تبادلگر تبلیغ^۱ اینترنتی به بررسی طیف گسترده‌ای از مشخصات فعلیت کاربران نامتعبر شامل رفتارهای مرتبط با کلیک‌های هرز پرداختند. روش دیگری که توسط دیو و همکارانش [۱۳] پیشنهاد شد، ابتدا کاربرانی را که بیشترین درآمد برای هر منشورکننده داشته‌اند، پیدا کند و سپس با مقایسه درآمد آنها با درآمد تولید شده از کاربران متناظر با منشورکننده‌گان درستکار، منشورکننده‌گان متقابل را شناسائی می‌نماید. در [۱۴]، مجموعه‌ای از محققان با کمک تکنیک‌های مختلف داده‌کاوی، الگوهای مشترک در بین کلیک‌های هرز را تشخیص می‌دهند. در پژوهشی مشابه، کیت و همکارانش [۱۵]، با تکیه بر تجزیه ۵ ساله خود در شبکه تبلیغات مایکروسافت، چالش‌های توسعه یک سیستم داده‌کاوی جهت شناسائی کلیک‌های هرز را بیان و پارامترهای مختلف در طراحی چنین سیستمی را تشریح می‌کنند. تبلیغات بلوف [۱۶] و ایجاد امضا برای بات‌ها [۱۷] از دیگر روش‌هایی هستند که با تشخیص کاربران معتبر، به مقابله با حملات کلیکی در شبکه‌های تبلیغاتی می‌پردازن. کارهای معرفی شده تاکنون، عمدهاً به مسئله شناسائی کلیک‌های هرز در شبکه‌های تبلیغاتی پرداخته‌اند، لذا مستقیماً به موضوع این مقاله مربوط نیستند. در ادامه تعدادی از روش‌هایی را که صرفاً به شناسائی کلیک‌های هرز در موتورهای جستجو (کلیک‌های هرز روی لینک‌های تبلیغاتی و نتایج معمولی) پرداخته‌اند و ما نیز در این مقاله از برخی ایده‌های آنها استفاده نموده‌ایم، مور می‌کنیم.

بو و همکارانش [۱۸]، سیستمی ارائه دادند که می‌تواند ترافیک‌های تولید شده توسط ربات‌ها را شناسائی نماید. آنها گروه‌هایی را از کاربران که حداقل یک پرس‌وجو/کلیک مشترک دارند، یافته و به کمک محاسبات ماتریسی به شbahat سنجی تاریخچه فعلیت‌های آنها می‌پردازن. این سیستم قادر به شناسائی ترافیک‌های غیر نرم‌الی که با نرخ پایین اما به صورت توزیع شده توسط شبکه‌ای از ربات‌ها ارسال می‌شود، می‌باشد. در [۱۹]، از تکنیک‌های یادگیری ماشین برای شناسائی کلیک‌های هرز استفاده می‌شود. به این ترتیب که ابتدا با به کارگیری کدهای کچا و تعدادی روش شهودی، حجم زیادی داده آموزشی برچسب‌دار تولید و سپس با توسعه یک الگوریتم نیمه نظرارتی از قدرت داده‌های فاقد برچسب برای بهبود کارایی دستبندی کننده استفاده می‌شود.

روش‌های جدیدتر با بررسی فعلیت کاربران در سطح نشست به شناسائی کلیک‌های هرز می‌پردازن. محققان در [۲۰]، هر نشست از کاربران را با کمک زنجیره مارکوف به صورت دنباله‌ای از جفت‌های (نوع فعلیت کاربر، شماره صفحه) مدل کرده و فاصله هر نشست از میانگین نشست‌ها را محاسبه می‌نمایند. فاصله زیاد بین‌گر نشست‌های غیر نرم‌الی بوده و تمام کلیک‌های انجام شده در آن به عنوان کلیک هرز در نظر گرفته می‌شود. در [۲۱] نیز نشست کاربران مدل می‌شود اما با توالی‌های سه تایی از (نوع فعلیت کاربر، هدف فعلیت و اختلاف زمانی هر فعلیت نسبت به فعلیت قبلی خود). بعد از مدل‌سازی نشست‌های کاربران، دو الگوریتم انتشار گراف دوبخشی "کاربر-نشست" و "نشست-الگو" برای شناسائی کلیک‌های هرز پیشنهاد می‌شود.

روش‌هایی که ذکر شد هر یک به جنبه خاصی از رفتارهای غیر نرم‌الی پرداخته و تنها قادر به شناسائی حملات انجام شده در آن دسته هستند.

آنها از دو روش ۱) نمایش تبلیغات مرتبه با پرس‌وجوی کاربر در کنار نتایج جستجو و ۲) نمایش تبلیغات گرافیکی در وبسایت‌های مرتبه با محتوای تبلیغ (یا اصطلاحاً منتشرکننده‌گان) بهره می‌گیرند. شبکه‌های تبلیغاتی معمولاً بر مبنای مدل "پرداخت به ازای هر کلیک" [۷] فعالیت می‌نمایند یعنی هر زمان روی یک تبلیغ کلیک شود، مبلغی از شارژ تبلیغ کننده کسر می‌گردد. در حالت دوم، موتورهای جستجو بخشی از درآمد هر کلیک را به منتشرکننده‌ای که کلیک از طریق آن انجام گرفته است، می‌دهند.

در سیستم تبلیغات آنلاین، کلیک‌های هرز می‌تواند توسط منتشرکننده‌گان و یا تبلیغ کننده‌گان انجام گیرد. منتشرکننده‌گان متقابل با انگیزه کسب درآمد بیشتر و همچنین تبلیغ کننده‌گان با هدف تمام کردن بودجه تبلیغات رقیب خود به این کار مبادرت می‌ورزند. گرچه با وجود این کلیک‌ها، موتورهای جستجو باز هم به ازای هر کلیک، درآمد خود را کسب می‌کنند اما تووجهی به این مسئله در بلندمدت موجب از بین رفت انتبار آنها نزد تبلیغ کننده‌گان می‌شود. بنابراین، شناسائی این نوع از کلیک‌ها و کسر نکردن شارژ تبلیغ کننده‌گان به ازای آنها بسیار مهم می‌باشد.

هرچند مسئله کلیک‌های هرز در سایر سیستم‌های آنلاین نظریه سیستم‌های

توصیه‌گر و شبکه‌های اجتماعی که از اطلاعات کلیک کاربران در تصمیم‌گیری‌های خود استفاده می‌کنند، وجود دارد، اما در مورد موتورهای جستجو به دلیل حجم

زیاد کاربران و ترافیک بسیار بالای آنها از اهمیت بسزایی برخوردار است. لذا ما در

این مقاله، صرفاً بر روی مسئله شناسائی کلیک‌های هرز در موتورهای جستجو

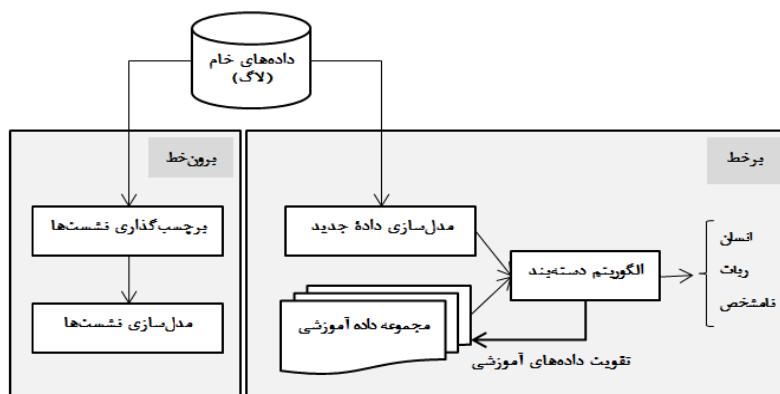
تمرکز می‌نماییم.

کلیک‌های هرز در موتورهای جستجو براساس اهدافشان به دو نوع مختلف تقسیم‌بندی می‌شوند: ۱) کلیک‌های هرز روی نتایج اصلی موتورهای جستجو با هدف افزایش تبیه یک وبسایت در صفحه نتایج و ۲) کلیک‌های هرز روی لینک‌های تبلیغاتی موجود در صفحه نتایج با هدف تمام کردن بودجه یک تبلیغ کننده خاص. موتورهای جستجو باید صرف نظر از نوع کلیک‌های هرز، آنها را شناسائی و از ترافیک کاربران نرمال تمایز دهند.

در گذشته، معمولاً حمله کننده‌گان از تعداد ثابتی آدرس IP برای تولید ترافیک غیر نرم‌الی استفاده می‌کردند، لذا شناسائی آنها نسبتاً ساده بود. اما به تدریج ابزارهای آنها پیش‌رفت کرد به نحوی که اکثر حملات امروزی به صورت کاملاً خودکار و توزیع شده توسط شبکه‌ای از ربات‌ها و بدافزارها انجام می‌گیرد [۱۱-۸]. بنابراین شناسائی آنها بسیار دشوار و پیچیده شده است. به عنوان مثال، در [۱۰] بدافزاری تشریح شده است که توانست روی بیش از ۴ میلیون کامپیوتر بنشیند و با تغییر آدرس "سرور نام میزبان"^۲ آنها به نمایش لینک‌های تبلیغاتی و انجام کلیک‌های هرز روی آنها بپردازد. این بدافزار به مدت ۴ سال ناشناخته ماند و ۱۴ میلیون دلار برای صاحبان خود به ارمغان آورد. لذا، به دلیل اهمیت موضوع، محققان زیادی از سراسر جهان به موضوع شناسائی ترافیک‌های غیر نرم‌الی و تفکیک آنها از ترافیک کاربران واقعی روی آورند.

در این مقاله، ما روش جدیدی مبتنی بر دسته‌بندی برای مقابله با کلیک‌های هرز، ارائه می‌کنیم. روش پیشنهادی شامل ۱) مدل‌سازی نشست‌های کاربران، ۲) دسته‌بندی نشست‌های مدل شده به دو دسته "نرم‌الی" و "غیر نرم‌الی" و ۳) به روزسانی و تقویت مجموعه آموزشی می‌باشد. در این پژوهش، برخلاف کارهای قلی، ما با ترکیب سه سطح از ویژگی‌ها (نشست، کاربر و آدرس IP) سعی نمودیم تمامی جنبه‌های رفتاری غیر نرم‌الی را پوشش داده و از آنها برای تفکیک کاربران انسانی و ربات‌ها بهره بگیریم. یکی دیگر از نقاط ممتاز این پژوهش، راهکار "تقویت مجموعه آموزشی" می‌باشد که به کمک آن می‌توان ضمن ایجاد پویایی در مجموعه داده‌های آموزشی، دقت دسته‌بندی را نیز افزایش داد.

ساختار مقاله در ادامه به صورت زیر می‌باشد: ابتدا کارهای مرتبه با موضوع پژوهش را در بخش دوم، مور می‌نماییم. در بخش سوم، سیستم پیشنهادی که شامل مجموعه ویژگی‌های استفاده شده و الگوریتم دسته‌بندی پیشنهاد شده جهت



شکل ۱- شمای کلی از سیستم پیشنهادی

نرمال بودن آن تصمیم‌گیری می‌شود.

مجموعه ویژگی‌های استخراج شده از داده به سه سطح تقسیم می‌شوند:
 ۱) ویژگی‌های رفتاری سطح نشست (2) ویژگی‌های رفتاری سطح کاربر و
 ۳) ویژگی‌های رفتاری سطح IP که به ترتیب به معنی هر یک می‌برداریم.

۱-۱-۱- ویژگی‌های سطح نشست

برای محاسبه ویژگی‌های سطح نشست، ما فعالیت‌های کاربر در هر نشست را مورد بررسی قرار می‌دهیم. زمانی که یک کاربر وارد موتور جستجو می‌شود، یک شناسه یکتا به عنوان "شناسه نشست" به او تخصیص داده می‌شود. این شناسه بعد از گذشت یک محدودیت زمانی (معمولًاً ۳۰ دقیقه) از فعل نبودن کاربر منقضی می‌شود و وقتی آن کاربر مجددًاً به سایت برگشت، یک شناسه نشست جدید به او داده می‌شود. هر کاربر ممکن است در طول نشست فعالیت‌های مختلفی در سیستم انجام دهد: به ارسال پرس‌وجو پردازد، صفحه نتایج را مرور نماید، روی لینک‌های نتایج کلیک کند، روی یک صفحه خاصی از مجموعه صفحات نتایج کلیک کند، پرس‌وجو خود را اصلاح نماید و غیره. در این مقاله ما تنها سه نوع فعالیت را در نظر می‌گیریم:

- ارسال یک پرس‌وجو (که آن به معنای پرس‌وجوهای مختلف می‌باشد).
 - برای مثال، Q_1 بیانگر یک پرس‌وجو است و Q_2 بیانگر یک پرس‌وجوی دیگر.
 - W_i : کلیک روی لینک نتایج جستجو یا کلیک روی لینک‌های موجود در صفحه نخست (که آن به معنای لینک‌های متفاوت می‌باشد).
 - N : کلیک روی شماره صفحات مختلف از مجموعه صفحات نتایج. این فعالیت می‌تواند شامل کلیک روی دکمه "صفحه بعد"، "صفحه قبل" و یا یک شماره صفحه خاص باشد.
- برای بررسی رفتار کاربر در هر نشست، مجموعه ویژگی‌های زیر از نشست حاری او استخراج می‌شود:

- احتمال مارکوف دنباله فعالیت‌های کاربر در نشست: این ویژگی در [۲۰] مطرح شده و مقدار آن پس از مدل‌سازی فعالیت‌های کاربر به صورت زنجیره مارکوف، از حاصل ضرب احتمال انتقال از یک وضعیت به وضعیت بعدی به دست می‌آید.

- تعداد کل فعالیت‌های کاربر در نشست برحسب (Q, W, N)
- تعداد کل کلیک‌های انجام شده روی نتایج وب
- تعداد کل پرس‌وجوهای ارسال شده
- تعداد کل کلیک‌های انجام شده روی شماره صفحات دیگر
- نسبت لینک‌های یکتای کلیک شده به کل لینک‌های کلیک شده
- نسبت دامنه‌های کلیک شده به کل لینک‌های کلیک شده

در این مقاله، ما جنبه‌های مختلفی از رفتار ترافیک‌های غیر نرمال را به صورت مجموعه‌ای از ویژگی‌ها در سه سطح نشست، کاربر و IP با یکدیگر ترکیب نموده و با کمک تکنیک دسته‌بندی پیشنهاد شده که گونه‌ای تغییر یافته از الگوریتم K-NN نزدیک‌ترین همسایه می‌باشد، به شناسایی کلیک‌های هرز می‌پردازم. همچنین بیشتر روش‌های قبلي به صورت آفلاین کار می‌کرند اما سیستم مطرح شده در این مقاله، قابلیت به کارگیری برخط را نیز دارا می‌باشد گرچه این قابلیت در اینجا مورد ارزیابی قرار نگرفته است.

۳- روش پیشنهادی

همانطور که در بخش قبل اشاره شد، روش‌هایی که اخیراً در زمینه شناسایی کلیک‌های هرز مطرح شده‌اند، عموماً به شناسایی نشست‌های غیر نرمال مตکی هستند. سیستمی که ما نیز در این مقاله پیشنهاد می‌دهیم در این دسته قرار می‌گیرد، با این تفاوت که با افزودن ویژگی‌هایی در سطح کاربر و آدرس IP سعی می‌کنیم دامنه تشخیص خود را گسترشده تر نموده تا بتوانیم رفتارهای مختلف غیر نرمال و ترافیک رباتی را با کمک این سیستم شناسایی نماییم.

شمای کلی از سیستم پیشنهاد شده در شکل ۱ نشان داده شده است. ما در این مقاله، از تکنیک دسته‌بندی نشست‌های کاربران بهره می‌گیریم. بنابراین روش مطرح شده شامل دو بخش آموزش و آزمایش (یا به عبارت دیگر برونو خط و برخط) می‌باشد. در فاز آموزش، نخست به معنی ویژگی‌ها و مدل‌سازی داده‌ها خواهیم پرداخت. سپس، چالش‌های موجود جهت تولید مجموعه داده آموزشی برچسب‌دار را مطرح نموده و با مرور تعدادی از روش‌های عنوان شده، روش مناسب خود را انتخاب می‌نماییم. در فاز آزمایش، الگوریتم دسته‌بندی پیشنهادی که گونه‌ای تغییر یافته از الگوریتم K-NN نزدیک‌ترین همسایه می‌باشد را معرفی می‌کنیم. در انتهای، تکنیکی جهت تقویت مجموعه داده آموزشی و افزایش دقت دسته‌بندی ارائه می‌دهیم.

۳-۱- مدل‌سازی داده‌ها

داده‌های استفاده شده در این پژوهش، لگ جستجوی یکی از پربازدیدترین موتورهای جستجوی فارسی (به آدرس parsijoo.ir) می‌باشد. هر رکورد از این مجموعه لگ، یک درخواست از سوی کاربر است که حاوی اطلاعات زمانی، پرس‌وجوهای ارسال شده، لینک کلیک شده، آدرس IP و شناسه کاربر می‌باشد. درخواست‌ها به ترتیب زمان واقعی‌شان پیمایش شده و به ازای هر درخواست مجموعه‌ای از ویژگی‌ها محاسبه شده و پس از دسته‌بندی در مورد نرمال یا غیر

[۲۳] ایده استفاده از کدهای کپچای کلیک‌پذیر مطرح شده است. با به کارگیری این نوع از کدهای کپچا کاربران کمتر به زحمت می‌افتدند و می‌توانند با سرعت و دقت آن را پشت سر بگذارند. اما همان‌طور که گفته شد، ایده نمایش کپچا به تمام کاربران موتور جستجو ایده خوبی نیست. کنگ و همکاران [۱۹]. برای تولید مجموعه داده آموزشی، پیشنهاد نمایش کپچا تنها به بخش کوچکی از کاربران را مطرح کردند. در این روش، تنها به کاربرانی که از آستانه‌های شهودی تعریف شده برای چند پارامتر ساده عبور می‌کنند، کدهای کپچا نمایش داده می‌شود. نویسنده‌گان مقاله ادعا کردند که با این روش، تنها از ۱٪ از کاربران خواسته شده که به کپچا پاسخ دهند که اکثر آن‌ها نیز به آن جواب نداده‌اند که این به معنای برنامه‌های رباتی می‌باشد که قادر به حل کپچا نیستند.

ما نیز مشابه [۱۹]، با استفاده از تعدادی روش شهودی ساده نظیر حجم فعالیت‌های کاربر در یک بازه زمانی کوتاه و لیست سیاهی از آدرس‌های IP، مجموعه اولیه‌ای از نشستهای غیر نرمال تولید نمودیم. در این حالت، اگر رفتار یک کاربر از آستانه‌های تعریف شده فراتر رفت، کد کپچا به سوی او ارسال می‌گردد. کاربر ممکن است به کد کپچا پاسخ ندهد، به اشتباہ پاسخ دهد و یا به درستی پاسخ دهد. در دو وضعیت نخست، ما نشست کاربر را به عنوان "مثبت" یا "ربات" برچست می‌زنیم. به این ترتیب، مجموعه‌ای از نشستهای غیر نرمال تولید می‌شود. در نهایت، ویژگی‌های سطح نشست، کاربر و آدرس IP به ازای آنها محاسبه شده (طبق بخش ۱-۳) و از آنها به عنوان مجموعه داده آموزشی اولیه در فرآیند دسته‌بندی استفاده می‌گردد. اما با روشی که در ادامه خواهیم گفت به تدبیح، به بهبود کیفیت مجموعه آموزشی کمک کرده و می‌توانیم با دقت بالاتری نسبت به دسته‌بندی نشستهای اقدام نماییم.

برای تولید مجموعه داده گفته شده، ما از دو هفته‌lag جستجو و کلیک استفاده نمودیم (از ۹۰/۰۹/۱۰ تا ۹۴/۰۹/۱۶) که شامل بیش از دو میلیون و سیصد هزار درخواست (پرس‌وجوها و کلیک‌ها) و بیش از نهصد و سی هزار نشست یکتا بود.

۳-۳- الگوریتم دسته‌بندی

در روش دسته‌بندی مطرح شده ما از الگوریتم K-نزدیک‌ترین همسایه^۳ (KNN) [۲۴]، به عنوان الگوریتم پایه استفاده کرده و سپس بنا به ضرورت، تعییراتی در آن لحاظ نمودیم. الگوریتم K-نزدیک‌ترین همسایه یک روش غیر پارامتری می‌باشد که به دلیل سادگی، سرعت و کارایی در بسیاری از مسائل دسته‌بندی و رگرسیون به عنوان مناسب‌ترین روش مورد استفاده قرار می‌گیرد. این الگوریتم برای دسته‌بندی یک داده جدید (داده آزمایشی)، آن را با کلیه نمونه‌های موجود در مجموعه داده آموزشی مقایسه و K-نزدیک‌ترین نمونه به آن را استخراج کرده و براساس برتری دسته یا برجسب مربوط به آنها، در مورد دسته داده آزمایشی مذبور تصمیم‌گیری می‌نماید. روش دسته‌بندی KNN در کنار سادگی، دارای دو مشکل اساسی می‌باشد: ۱) حافظه مصرفی و ۲) حجم زیاد محاسبات پردازشی. اولی به دلیل نگه‌داری کل مجموعه داده آموزشی در حافظه و دومی به علت انجام محاسبه فاصله داده جدید با کلیه نمونه‌های آموزشی به وجود می‌آید. هرچه مجموعه داده آموزشی بزرگ‌تر شود، دو مشکل گفته شده بیشتر خود را نشان می‌دهند. ما سعی می‌کنیم در این مقاله با ارائه ایده‌هایی بر دو مشکل فوق فائق آییم و الگوریتم را برای کاربردان مناسب‌سازی نماییم.

در این پژوهش، ما از الگوریتم K-نزدیک‌ترین همسایه تک کلاسه استفاده می‌کنیم. مهم‌ترین دلیل انتخاب این گونه، کاهش حجم داده آموزشی از طریق نگه‌داری تنها نمونه‌های مثبت (غیر نرمال) می‌باشد. در ادامه، ابتدا معرفی به الگوریتم K-نزدیک‌ترین همسایه تک کلاسه نموده و سپس گونه تغییر یافته از آن را پیشنهاد می‌دهیم.

- نسبت پرس‌وجوهای یکتای ارسال شده به کل پرس‌وجوهای ارسال شده
- نسبت مجموع فعالیت‌های یکتای انجام شده به کل فعالیت‌ها
- نسبت پرس‌وجوهای ارسال شده به مدت زمان فعالیت نشست
- نسبت کلیک‌های انجام شده به مدت زمان فعالیت نشست
- نسبت کلیک‌های روی شماره صفحات دیگر به مدت زمان فعالیت نشست هفت ویژگی آخر برای اولین بار در این مقاله پیشنهاد شده‌اند و مابقی در [۲۰] به کار رفته‌اند. مقادیر تمام ویژگی‌های فوق و همین‌طور ویژگی‌هایی که در ادامه خواهند آمد به بازه [۱۰-۰] نرمال‌سازی می‌شوند.

۳-۱-۲- ویژگی‌های سطح کاربر

ویژگی‌های مطرح شده در سطح نشست، از نشست جاری کاربر محاسبه می‌گردد. اما هر کاربر ممکن است تاکنون نشستهای متعددی داشته باشد. بنابراین می‌توان تمام تاریخچه نشستهای کاربر را در قالب ویژگی‌های سطح کاربر لحاظ کرد. ویژگی‌های مطرح شده در سطح کاربر، عیناً مانند ویژگی‌های سطح نشست هستند با این تفاوت که از میانگین کلیه نشستهای کاربر محاسبه می‌گردد.

۳-۱-۳- ویژگی‌های سطح IP

بسیاری از ربات‌ها قادر به اجرای کدهای جاوا اسکریپت نیستند و یا کوکی آنها غیر فعال است. بنابراین به ازای هر درخواست که از جانب آنها به موتور جستجو می‌آید یک شناسه کاربری جدید به آنها تخصیص می‌یابد، در نتیجه هر نشست از آنها تنها شامل یک فعالیت است. بنابراین ویژگی‌های سطح نشست و سطح کاربر به تنها یک کافی نیستند بلکه نیاز به وجود ویژگی‌هایی در سطح IP نیز می‌باشد. ویژگی‌های که ما برای هر IP در نظر گرفتیم عبارتند از:

- آدرس IP اینترنت یا اینترنت (0/0): کاربرانی که از سازمان‌ها و ادارات داخل کشوری هستند دارای آدرس IP اینترنت می‌باشند.
 - تعداد کل فعالیت‌های IP
 - تعداد پرس‌وجوهای ارسال شده
 - تعداد کل کلیک‌های انجام شده روی نتایج جستجو
 - تعداد کل کلیک‌های انجام شده روی شماره صفحات
 - نسبت لینک‌های یکتای کلیک شده به کل لینک‌های کلیک شده
 - نسبت دامنه‌های یکتای کلیک شده به کل لینک‌های کلیک شده
 - نسبت کوکی‌های تخصیص یافته به کل فعالیت‌های هر IP
- هر نشست از کاربر ممکن است با یک یا چند آدرس IP همراه باشد بنابراین ویژگی‌های فوق به صورت میانگین روی تمامی آدرس‌های آن نشست محاسبه می‌گردد.

۳-۲- تولید مجموعه داده آموزشی اولیه

حجم بالای lag موتورهای جستجو، برچسب‌گذاری دستی آنها به عنوان "انسان‌اربات" جهت تولید مجموعه داده آموزشی را تقریباً ناممکن می‌سازد. در برخی از سرویس‌های اینترنتی مانند ایمیل یا بانکداری الکترونیک می‌توان برای دسترسی به خدمات از مکانیزم کپچا جهت احراز هویت کاربران و تمایز آنها از ربات‌ها استفاده نمود. به این ترتیب تنها کاربران واقعی امکان دسترسی به سرویس را خواهند داشت و فعالیت ربات‌ها پشت کپچا متوقف می‌شود. اما این روش برای موتورهای جستجو کاربردی نیست زیرا هدف موتورهای جستجو، سرعت و سهولت در ارائه خدمات با کمترین فعالیت اضافه از سوی کاربر می‌باشد. با این وجود، در

۴-۳- تقویت مجموعه داده آموزشی

همان طور که گفته شد، ما یک مجموعه اولیه‌ای از نمونه‌های آموزشی تولید نمودیم اما برای افزایش دقت دسته‌بندی، نیاز به افزودن نمونه‌های بیشتر به مجموعه داده آموزشی داریم. از طرف دیگر، افزودن نمونه‌های بیشتر به معنی مصرف حافظه و محاسبات بیشتر در زمان دسته‌بندی می‌باشد. از همین رو، یک فیلد جدید به نام شمارنده (با مقدار اولیه ۱) به هر نمونه آموزشی اضافه نمودیم. در زمان دسته‌بندی، اگر داده آزمایشی به عنوان نمونه "مثبت" برچسب زده شد، به ازای تمام نمونه‌های موجود در مجموعه همسایگی داده آزمایشی، چنانچه فاصله محاسبه می‌گردد. برای این منظور از معیار فاصله اقلیدسی که متدالوں ترین معیار فاصله است، استفاده می‌گردد:

$$c_i = c_i + \frac{d(x', x_i^{NN})}{\sum_{j=1}^K d(x', x_j^{NN})} \quad (5)$$

که d بیانگر فاصله داده آزمایشی از نمونه آموزشی می‌باشد. هرچه مقدار فیلد شمارنده یک داده آموزشی بزرگتر باشد، به این معنا است که این داده به نمایندگی از تعداد بیشتری داده در مجموعه آموزشی حضور دارد و لذا از اهمیت بالاتری برخوردار است. در نتیجه طبق رابطه (۴) سهم بیشتری در تعیین برچسب یک داده جدید خواهد داشت. به همین ترتیب، نمونه‌های آموزشی با مقدار شمارنده کوچکتر، نقش کمتری را در تعیین برچسب داده جدید ایفا می‌کنند. مقدار α به صورت تجربی برابر با ۰.۲ در نظر گرفته شده است. نتایج حاصل از افزوده شدن این فیلد و بهبود کارایی الگوریتم دسته‌بندی در بخش بعد نشان داده خواهد شد.

سیستم مطرح شده در این مقاله، قابلیت به کارگیری به صورت برخط را دارا می‌باشد. در سیستم برخط، هرگاه موردی به عنوان غیر نرمال تشخیص داده شود، منجر به نمایش کپچا خواهد شد. در این شرایط، پیشنهاد می‌شود مواردی که منجر به نمایش کپچا می‌شوند و از طرف دیگر کاربر به آنها پاسخ نمی‌دهد (ربات‌ها)، به مجموعه آموزشی اضافه گردد. برای این منظور لازم است ابتدا به دلیل محدودیت حافظه و سریار پردازشی در زمان دسته‌بندی، آستانه‌ای برای تعداد داده‌های آموزشی در نظر گرفته شود. سپس می‌توان آن داده را تحت دو شرط زیر به مجموعه آموزشی اضافه نمود:

(۱) اگر اندازه مجموعه آموزشی کمتر از محدودیت تعیین شده باشد، داده آزمایشی به مجموعه آموزشی افزوده می‌شود.

(۲) در غیر این صورت دو نمونه آموزشی از مجموعه نمونه‌های آموزشی که برچسب یکسانی داشته و کمترین فاصله را از یکدیگر دارند یافته و آنها را با هم ادغام می‌کنیم تا فضا برای نگهداری نمونه آزمایشی جدید باز شود. برای ادغام دو نمونه آموزشی میانگین بردار ویژگی‌های آن دو را محاسبه و فیلدی‌های شمارنده آنها را با هم جمع کنیم، یعنی دو نمونه (x_i, c_i) و (x_j, c_j) را از مجموعه داده آموزشی حذف و نمونه جدید $(x_i + x_j, c_i + c_j)$ و همچنین نمونه آزمایشی را به مجموعه آموزشی اضافه می‌نماییم.

۴- ارزیابی

گرچه هریک از روش‌های عنوان شده در بخش ۲، با رویکرد متفاوتی به مقابله با کلیک‌های هر زمینه پردازند اما عدم وجود مجموعه داده عمومی از نشسته‌های نرمال و غیر نرمال از رفتار کاربران، موجب دشوار شدن عملیات ارزیابی و مقایسه این روش‌ها می‌شود. لذا عموماً ارزیابی‌ها به بررسی نمونه‌های غیر نرمال توسط افراد

اگر مجموعه داده‌های آموزشی را به صورت $\{x_1, x_2, \dots, x_N\}$ در نظر بگیریم به نحوی که x_i نمونه آموزشی i -ام در فضای m بعدی باشد، برچسب نمونه آزمایشی x' در دو گام مشخص می‌گردد:

ابتدا فاصله نمونه آزمایشی از تمام نمونه‌های موجود در مجموعه آموزشی محاسبه می‌گردد. برای این منظور از معیار فاصله اقلیدسی که متدالوں ترین معیار فاصله است، استفاده می‌گردد:

$$d(x', x_i) = \|x' - x_i\|_{L_2} \quad (1)$$

منظور از L_2 یک بردار، مجذور مجموع مربعات مقادیر آن بردار می‌باشد. K نمونه آموزشی که کمترین فاصله تا داده آزمایشی داشته باشند در مجموعه همسایگی آن قرار می‌گیرند که این مجموعه همسایگی را با NN نمایش می‌دهیم. فرض کنید x نزدیک‌ترین همسایه به x' در مجموعه داده آموزشی باشد. داده آزمایشی به کلاس مثبت تعلق می‌گیرد اگر:

$$\frac{\sum_{i=1}^K d(x', x_i^{NN})}{\sum_{i=1}^K d(x_i^{NN}, \text{near}(x_i^{NN}))} < \delta \quad (2)$$

که d فاصله داده آزمایشی از i -امین داده موجود در مجموعه همسایگی اش و $\text{near}(x_i^{NN})$ فاصله i -امین داده موجود در مجموعه همسایگی داده آزمایشی تا نزدیک‌ترین همسایه به خودش در مجموعه داده‌های آموزشی می‌باشد. نسبت گفته شده باید از آستانه δ کمتر باشد تا داده آزمایشی برچسب "مثبت" دریافت کند (معمولًا $\delta = 0.5$ در نظر گرفته می‌شود).

در این مقاله، ما به هر داده آموزشی یک فیلد شمارنده نیز اضافه می‌نماییم، یعنی $\{c_1, c_2, \dots, c_N\}$. هدف از افزودن این فیلد آن است که هر داده آموزشی بتواند به نمایندگی از چندین نقطه در مجموعه داده آموزشی حضور داشته باشد، بنابراین می‌توان بنا بر ظرفیت حافظه، یک محدودیت برای تعداد داده‌های آموزشی در نظر گرفت و از این طریق به مشکلات حافظه مصرفی و حجم محاسبات فائق شد و از طرف دیگر کیفیت داده‌های آموزشی را نیز افزایش داد تا دقت فرآیند دسته‌بندی افزایش یابد. برای این منظور ابتدا رابطه (۲) را به صورت زیر بازنویسی می‌کنیم:

$$\frac{\sum_{i=1}^K \frac{1}{d(x', x_i^{NN})}}{\sum_{i=1}^K \frac{1}{d(x_i^{NN}, \text{near}(x_i^{NN}))}} > \delta' \quad (3)$$

که در این رابطه، صورت کسر بیانگر میزان شباهت داده آزمایشی به مجموعه همسایگی اش و مخرج کسر میزان نزدیکی نمونه‌های موجود در مجموعه همسایگی داده آزمایشی به سایر داده‌های آموزشی می‌باشد. سپس فیلد شمارنده c نیز به عنوان ضریب به آن اضافه می‌کنیم:

$$\frac{\sum_{i=1}^K \frac{c_i^{NN}}{d(x', x_i^{NN})}}{\sum_{i=1}^K \frac{c_i^{NN}}{d(x_i^{NN}, \text{near}(x_i^{NN}))}} > \delta' \quad (4)$$

که c_i^{NN} شمارنده متناظر با i -امین نمونه موجود در مجموعه همسایگی داده آزمایشی می‌باشد. روش به روز رسانی مقدار c_i ها در بخش بعدی خواهد آمد و همچنین مقدار آستانه δ' در بخش ۴ تعیین می‌شود.

سپس این مجموعه را به عنوان داده‌های آزمایشی به سیستم دسته‌بند تک کلاسه تزریق نمودیم. پس از دسته‌بندی داده‌های این مجموعه و به روز رسانی فیلد شمارنده نمونه‌های آموزشی، ما مجددًا مجموعه آموزشی تقویت شده را با کمک روش اعتبارسنجی متقابل ۱۰ وجهی مورد ارزیابی قرار دادیم. نتایج حاصل در جدول ۲ نمایش داده شده است.

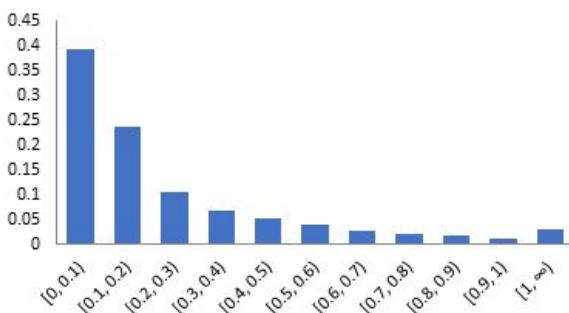
جدول ۲- نتایج اعتبارسنجی متقابل ۱۰ وجهی روی مجموعه داده تقویت شده

K = 7	K = 6	K = 5	K = 4	K = 3	K = 2	K = 1	
۴۹۴۷	۴۹۴۶	۴۹۴۴	۴۹۳۰	۴۹۲۴	۴۹۰۹	۴۸۸۲	مثبت صحیح
۵۳	۵۴	۵۶	۷۰	۷۶	۹۱	۱۱۸	منفی کاذب
۹۸/۹۴	۹۸/۹۲	۹۸/۸۸	۹۸/۶	۹۸/۴۸	۹۸/۱۹	۹۷/۶۴	دقت (%)

مقایسه نتایج جدول ۱ و ۲، ضمن اینکه بر بھبود دسته‌بند پیشنهادی با افزودن فیلد شمارنده و تقویت داده‌های آموزشی دلالت دارد (افزایش دقت تا ۰٪/۲)، نشان می‌دهد که دقت دسته‌بند پس از تقویت داده‌های آموزشی، به ازای مقادیر مختلف اندازه همسایگی در حدود ٪۹۸ ثابت باقی می‌ماند، لذا ما با توجه به نتایج جدول ۱، از مقدار K=5 برای اندازه مجموعه همسایگی در الگوریتم KNN استفاده نمودیم، بنابراین با این روش، دقت روش پیشنهادی برابر با ٪۹۶/۵۴ خواهد بود.

۴-۲- کارایی الگوریتم دسته‌بندی

در این بخش، کارایی الگوریتم پیشنهادی با استفاده از لاغ یک هفته (۹۴/۰۹/۱۷) تا (۹۴/۰۹/۲۵) که شامل بیش از ۲ میلیون رکورد بود، ارزیابی می‌شود. با توجه به الگوریتم پیشنهادی، به هر نشست پس از دسته‌بندی یک امتیاز تخصیص داده می‌شود (رابطه ۴). این امتیاز در بازه [۰ - ۱] متغیر است. ما ابتدا فرکانس امتیاز داده شده به هر نشست را در بازه‌های مختلف به دست می‌آوریم. طول همه بازه‌ها ۱/۰ در نظر گرفته شد ولی جهت محدود کردن تعداد بازه‌ها، امتیازات بزرگتر یا مساوی با ۱ را به صورت مجمع‌آوری در یک بازه در نظر می‌گیریم. فرکانس امتیازهای تخصیص داده شده به هر نشست در بازه‌های مختلف در شکل ۲ نشان داده شده است.



شکل ۲- فرکانس امتیازهای حاصل از دسته‌بندی در بازه‌های مختلف

با توجه به الگوریتم پیشنهادی، می‌دانیم هرچه امتیاز محاسبه شده بیشتر باشد، احتمال غیر نرمال بودن آن نشست بیشتر می‌شود. ما برای آزمایشات بیشتر، از بین نشست‌هایی که امتیاز بالاتر از ۵/۰ داشتند، به صورت تصادفی ۲۰۰ نشست

خبره و محاسبه معیارهایی نظیر دقت^۴ محدود می‌شود. ما نیز در این پژوهش، با همین محدودیت مواجه بودیم و تنها امکان محاسبه این معیار را داشتیم، با این حال سعی نمودیم دقت روش‌های پیشنهادی را از جنبه‌های مختلف مورد ارزیابی و مقایسه قرار دهیم.

در ادامه، ابتدا به ارزیابی مجموعه داده آموزشی و مقایسه مجموعه داده اولیه و مجموعه داده آموزشی تقویت شده می‌پردازیم. سپس نحوه انتخاب پارامترها و کارایی الگوریتم دسته‌بند را مورد ارزیابی قرار می‌دهیم و در نهایت دقت روش پیشنهادی را با یکی از آخرین و بهترین کارهای مرتبط مطرح شده مقایسه خواهیم نمود.

۴-۱- اعتبارسنجی متقابل k وجهی^۵

در ابتدا برای ارزیابی مجموعه داده آموزشی از روش اعتبارسنجی متقابل ۱۰ وجهی استفاده می‌نماییم. در این روش، نمونه‌های آموزشی به صورت تصادفی به ۱۰ بخش مساوی تقسیم می‌شوند، از یک بخش به عنوان داده ارزیابی و از نه بخش دیگر به عنوان داده آموزشی استفاده می‌شود. این فرآیند ۱۰ بار تکرار می‌شود، بنابراین از تمامی نمونه‌ها برای آموزش استفاده می‌شود و هر نمونه نیز یکبار برای ارزیابی مورد استفاده قرار می‌گیرد. در نهایت، مجموع/میانگین نتایج هر تکرار به عنوان تخمین نهایی محاسبه می‌گردد. روش اعتبارسنجی متقابل از معیار دقت دسته‌بندی^۶ بهره می‌گیرد:

$$CA = \frac{TP}{TP + FN} \quad (6)$$

که "مثبت صحیح"^۷ بیانگر تعداد نمونه‌هایی است که دسته واقعی آنها "مثبت" بوده و الگوریتم دسته‌بندی نیز دسته آنها را به درستی "مثبت" تشخیص داده است و "منفی کاذب"^۸ معرف تعداد نمونه‌هایی است که دسته واقعی آنها "مثبت" بوده و الگوریتم دسته‌بندی، دسته آنها را به اشتباه "منفی" تشخیص داده است. معیار دقت دسته‌بندی، بیانگر این حقیقت است که دسته‌بند طراحی شده قادر است تا چند درصد از کل مجموعه نمونه‌های آزمایشی را به درستی دسته‌بندی نماید. کمترین مقدار دقت یک دسته‌بند، صفر (ضعیفترین کارایی) و بیشترین مقدار آن، یک (بهترین کارایی) می‌باشد.

مجموعه داده آموزشی اولیه ما شامل ۵۰۰ نمونه مثبت که همگی دارای فیلد شمارندهای با مقدار "۱" بودند و الگوریتم فوق را برای مقادیر مختلف K (اندازه مجموعه همسایگی) در الگوریتم KNN تکرار نمودیم. نتایج آن در جدول ۱ نشان داده شده است.

جدول ۱- نتایج اعتبارسنجی متقابل ۱۰ روی مجموعه داده اولیه

K = 7	K = 6	K = 5	K = 4	K = 3	K = 2	K = 1	
۴۸۳۳	۴۸۳۱	۴۸۲۷	۴۷۹۷	۴۷۵۱	۴۷۲۱	۴۶۷۶	مثبت صحیح
۱۶۷	۱۶۹	۱۷۲	۲۰۳	۲۴۹	۲۷۹	۳۲۴	منفی کاذب
۹۶/۶۶	۹۶/۶۲	۹۶/۵۴	۹۵/۹۴	۹۵/۰۲	۹۴/۴۲	۹۳/۶۲	دقت (%)

در گام بعد، کیفیت داده‌های آموزشی را پس از به روز رسانی فیلد شمارنده آنها مورد بررسی قرار می‌دهیم. برای این منظور، از لاغ یک هفت‌های تراکنش کاربران (۱۷ تا ۹۴/۰۹/۲۵) استفاده نموده و آنها را مدل‌سازی نمودیم.

مقاله قادر است تا به صورت لحظه‌ای و با مصرف حافظه پائین‌تر و انجام محاسباتی ساده‌تر به شناسائی کلیک‌های هرز پردازد.

۵- نتیجه‌گیری

در این مقاله، یک روش جدید و کارآمد مبتنی بر دسته‌بندی جهت شناسائی نشست‌ها و کلیک‌های هرز ارائه نمودیم. ما در ابتدا ویژگی‌های مهمی را که به تمایز رفتار کاربران نرمال و ربات‌ها کمک می‌کردیم و سپس الگوریتم دسته‌بندی جدیدی که مبتنی بر الگوریتم KNN می‌باشد را پیشنهاد دادیم که در آن با افزودن یک پارامتر ساده "شمارنده" توانستیم بر مشکلات الگوریتم اولیه KNN که شامل حافظه مصرفی و حجم زیاد محاسبات بود، فائق آییم. اما از طرف دیگر با ایجاد مکانیزم بهروزرسانی مجموعه آموزشی، سعی نمودیم همواره امکان افزودن نمونه‌های جدید به سیستم وجود داشته باشد. در بخش ارزیابی نشان دادیم که الگوریتم پیشنهادی می‌تواند کلیک‌های هرز را با دقیقیت بیش از ۹۶٪ تشخیص دهد که در مقایسه با روش‌های قبلی بهبود مناسبی از خود نشان می‌دهد. ما قصد داریم در آینده‌ای نزدیک، الگوریتم مطرح شده را در محیط برخط آزمایش و نتایج آن را در کارهای بعدی گزارش نماییم.

تشکر و قدردانی

از مجموعه موتور جستجوی پارسی جو جهت تأمین داده‌های مورد نیاز در این پژوهش تشکر می‌نماییم.

مراجع

- [1] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford InfoLab, 1999.
- [2] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, "Link-Based Characterization and Detection of Web Spam," *Proc. Intl Workshop on Adversarial Information Retrieval on the Web*, 2006.
- [3] Y. Liu, R. Cen, M. Zhang, S. Ma, and L. Ru, "Identifying Web Spam with User Behavior Analysis," *Proc. Intl Workshop on Adversarial Information Retrieval on the Web*, pp. 9–16, 2008.
- [4] A. Karasardis, B. Rexroad, and D. Hoeflin, "Wide-scale Botnet Detection and Characterization," *Proc. Conf. First Workshop on Hot Topics in Understanding Botnets*, pp. 7, 2007.
- [5] Z. Dou, R. Song, X. Yuan, and J.-R. Wen, "Are Click-through Data Adequate for Learning Web Search Rankings?," *Proc. ACM Conf. Information and Knowledge Management*, pp. 73–82, 2008.
- [6] I. A. Board, "Internet Advertising Revenue Report," <https://www.iab.com>, 2015.
- [7] D. Szetela, and J. Kerschbaum, *Pay-Per-Click Search Engine Marketing: An Hour a Day*, USA: SYBEX Inc.,

را انتخاب و دقت الگوریتم پیشنهادی را مورد بررسی قرار دادیم که نتایج آن در جدول ۳ نمایش داده شده است.

جدول ۳- دقت الگوریتم پیشنهادی در بازه‌های امتیازی مختلف

حدوده امتیاز	نشست‌ها	تعداد غیر نرمال‌ها	دقت (%)
[۰.۵-۰.۶)	۴۱	۲	۴۱/۸۷
[۰.۶-۰.۷)	۲۳	۳	۱۳/۰۴
[۰.۷-۰.۸)	۱۸	۵	۲۷/۷۷
[۰.۸-۰.۹)	۱۹	۱۷	۸۹/۴۷
[۰.۹-۱)	۱۵	۱۴	۹۳/۳۳
[۱-۱۰)	۸۴	۸۳	۹۸/۸۰

با توجه به اینکه شناسائی موارد هرز نیاز به دقت بالایی دارد، لذا ما نشست‌هایی که امتیاز بالاتر از ۰/۸ داشته‌اند ($\delta = 0.8$) را به عنوان نشست غیر نرمال در نظر گرفتیم. با این حساب، دقت روش برابر با ۹۶/۶۱٪ می‌شود که از نسبت ۱۱۴ نشست غیر نرمال درست تشخیص داده شده به ۱۱۸ نشست مورد بررسی در بازه [۰/۸-۱۰/۰] به دست می‌آید. در نتیجه کلیه کلیک‌های انجام شده در این نشست‌ها به عنوان کلیک هرز قلمداد می‌شوند.

۴- مقایسه با کارهای قبلی

در نهایت، کارایی الگوریتم پیشنهادی با دو روش اخیر مطرح شده جهت شناسائی کلیک‌های هرز [۲۰، ۲۱] نیز مقایسه گردید. در روش معروف شده در [۲۰]، ابتدا نشست‌های کاربران به صورت توالی‌های دوتایی از (نوع فعالیت کاربر و شماره صفحه) و با کمک زنجیره مارکوف مدل می‌شود. سپس فاصله هر نشست از میانگین نشست‌ها با معیار فاصله ماهالانوبیس محاسبه می‌گردد. آنگاه نشست‌هایی که در یک درصد بالای توسعه فاصله نشست‌ها قرار دارند، به عنوان نشست غیر متعارف در نظر گرفته می‌شوند. همچنین، در الگوریتم دیگری که در سال ۲۰۱۴ معرفی شد [۲۱] مطرح گردید، ابتدا نشست‌های کاربران به صورت توالی‌های سه‌تایی از (نوع فعالیت، هدف فعالیت و اختلاف زمانی آن با فعالیت قبلی) مدل گردیده و سپس الگوریتم گراف دو بخشی کاربر-نشست برای شناسائی تعداد بیشتری از نشست‌های مشکوک اعمال می‌شود. در انتهای نشست‌هایی که امتیازی بالاتر از ۰/۹ داشته‌اند به عنوان نشست غیر نرمال در نظر گرفته می‌شوند. از بین نتایج حاصل از این دو روش به همراه روش پیشنهادی، به صورت تصادفی ۲۰۰ نمونه از آنها را انتخاب و درستی تشخیص آنها را بررسی نمودیم. نتایج دقت هر سه روش در جدول ۴ نشان داده شده است.

جدول ۴- مقایسه با روش‌های قبلی

دقت (%)	
۸۹	روش نشست‌های متعارف/غیر متعارف [۲۰]
۹۱.۵	روش گراف دوبخشی [۲۱]
۹۶/۶۱	روش پیشنهادی

مقایسه دقت محاسبه شده به ازای روش‌های قبلی و روش پیشنهادی، نشان می‌دهد که روش پیشنهاد شده در این مقاله در حدود ۵/۱ درصد عملکرد بهتری روی داده‌های ما داشته است. البته باید به این نکته نیز توجه نمود که روش‌های قبلی بیشتر به صورت برونو خط عمل می‌کنند، در حالیکه روش پیشنهادی در این

Graph Propagation," *Proc. ACM Intl Conf. Web Search and Data Mining*, pp. 93–102, 2014.

[22] G. Buehrer, J. W. Stokes, and K. Chellapilla, "A Large-scale Study of Automated Web Search Traffic," *Proc. Intl Workshop on Adversarial Information Retrieval on the Web*, pp. 1–8, 2008.

[23] R. A. Costa, R. J. G. B. de Queiroz, and E. R. Cavalcanti, "A Proposal to Prevent Click-Fraud Using Clickable CAPTCHAs," *Proc. IEEE Intl Conf. Software Security and Reliability Companion*, pp. 62–67, 2012.

[24] T. Cover, and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. Inf. Theor.*, vol. 13, no. 1, pp. 21–27, Sep. 2006.

مهندیه فلاح مدرک کارشناسی خود را در رشته مهندسی فناوری اطلاعات در دانشگاه صنعتی اصفهان اخذ نمود.
پس از آن در مقطع کارشناسی ارشد در همان رشته با گرایش شبکه‌های کامپیوتری به ادامه تحصیل پرداخت.
وی به موتورهای جستجو و شبکه‌های کامپیوتری علاقه زیادی دارد و هم‌اکنون در پارسی‌جو، اولین موتور جستجوی ایرانی مشغول به فعالیت می‌باشد.



آدرس پست‌الکترونیکی ایشان عبارت است از:
m.fallah@stu.yazd.ac.ir

سجاد ظریف‌زاده دکتری خود را در رشته مهندسی کامپیوتر از دانشگاه تهران در سال ۱۳۹۱ دریافت نمود.
وی هم‌اکنون استادیار و عضو هیئت علمی گروه مهندسی کامپیوتر دانشگاه یزد می‌باشد. زمینه‌های تحقیقاتی اصلی وی در مورد سرویس‌های مبتنی بر وب و همچنین سیستم‌ها و شبکه‌های کامپیوتری می‌باشد.



آدرس پست‌الکترونیکی ایشان عبارت است از:
szarifzadeh@yazd.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۵/۰۵/۱۲

تاریخ اصلاح: ۱۳۹۵/۰۶/۰۴

تاریخ قبول شدن: ۱۳۹۵/۰۶/۱۴

نویسنده مرتبه: مهندیه فلاح، دانشکده برق و کامپیوتر، دانشگاه یزد، بیزد، ایران.

2010.

[8] N. Daswani, and M. Stoppelman, "The Anatomy of Clickbot.A," *Proc. Conf. First Workshop on Hot Topics in Understanding Botnets*, pp. 11, 2007.

[9] B. Miller, P. Pearce, C. Grier, C. Kreibich, and V. Paxson, "What's Clicking What? Techniques and Innovations of Today's Clickbots," *Proc. Intl Conf. Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 164–183, 2011.

[10] S. A. Alrwais, A. Gerber, C. W. Dunn, O. Spatscheck, M. Gupta, and E. Osterweil, "Dissecting Ghost Clicks: Ad Fraud via Misdirected Human Clicks," *Proc. Conf. Computer Security Applications*, pp. 21–30, 2012.

[11] P. Pearce, and et. al., "Characterizing Large-Scale Click Fraud in ZeroAccess," *Proc. ACM SIGSAC Conf. Computer and Communications Security*, pp. 141–152, 2014.

[12] B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna, "Understanding Fraudulent Activities in Online Ad Exchanges," *Proc. ACM SIGCOMM Conf. Internet Measurement*, pp. 279–294, 2011.

[13] V. Dave, S. Guha, and Y. Zhang, "ViceROI: Catching Click-spam in Search Ad Networks," *Proc. ACM SIGSAC Conf. Computer and Communications Security*, pp. 765–776, 2013.

[14] R. Oentaryo, and et. al., "Detecting Click Fraud in Online Advertising: A Data Mining Approach," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 99–140, Jan. 2014.

[15] B. Kitts, J. Y. Zhang, G. Wu, W. Brandi, J. Beasley, K. Morrill, J. Ettedgui, S. Siddhartha, H. Yuan, F. Gao, P. Azo, and R. Mahato, "Click Fraud Detection: Adversarial Pattern Recognition over 5 Years at Microsoft," *Springer Intl Pub. Real World Data Mini. Apps.*, vol. 17, pp. 181–201, 2015.

[16] H. Haddadi, "Fighting Online Click-fraud Using Bluff Ads," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 2, pp. 21–25, Apr. 2010.

[17] B. Kitts, J. Y. Zhang, A. Roux, and R. Mills, "Click Fraud Detection with Bot Signatures," *Proc. IEEE Intl Conf. Intelligence and Security Informatics*, pp. 146–150, 2013.

[18] F. Yu, Y. Xie, and Q. Ke, "SBotMiner: Large Scale Search Bot Detection," *Proc. ACM Intl Conf. Web Search and Data Mining*, pp. 421–430, 2010.

[19] H. Kang, K. Wang, D. Soukal, F. Behr, and Z. Zheng, "Large-scale Bot Detection for Search Engines," *Proc. Intl Conf. World Wide Web*, pp. 501–510, 2010.

[20] N. Sadagopan, and J. Li, "Characterizing Typical and Atypical User Sessions in Clickstreams," *Proc. Intl Conf. World Wide Web*, pp. 885–894, 2008.

[21] X. Li, M. Zhang, Y. Liu, S. Ma, Y. Jin, and L. Ru, "Search Engine Click Spam Detection Based on Bipartite

¹ Domain Name Server

² Ad Exchanger

³ K-Nearest Neighbor (KNN)

⁴ Precision

⁵ K-Fold Cross-Validation

⁶ Classification Accuracy

⁷ True Positive (TP)

⁸ False Negative (FN)

طراحی و پیاده‌سازی جمع‌کننده دهدۀ افزونه‌ای با توان مصرفی پایین

لیلا میرمقتدایی

سعید گرگین

پژوهشکده برق و کامپیوتر، سازمان پژوهش‌های علمی و صنعتی ایران، تهران، ایران

چکیده

در سیستم‌های کامپیوتراً عمل جمع جزء اصلی کلیه پردازش‌ها و پایه تمام عملیات حسابی است؛ به طوری که افزایش سرعت در عمل جمع بر کارآیی کل سیستم تاثیر شگرفی دارد. در نظام‌های عددی متغّر، انتشار نقلی باعث واستگی زمان جمع به طول عملوندها می‌شود که این مسئله در عملیات با دقت بالا مشکل آفرین است و تاخیر زیادی را تحمیل خواهد کرد. حال آنکه با به کارگیری نظام‌های عددی نامتغّر افزونه‌ای، عمل جمع بدون انتشار نقلی و در زمان ثابت قابل انجام است. در این مقاله، با توجه به اهمیت روزافزون سیستم‌های محاسباتی مبنای ده، با استفاده از الگوریتم افزایی وزن دار و خاصیت افزونگی ذاتی موجود در ارقام دهدۀ، تقسیم‌بندی جدیدی برای جمع اعداد دهدۀ افزونه‌ای ارائه شده است که با حفظ سرعت، توان مصرفی و مساحت روی تراشه را نسبت به روش‌های پیشین کاهش می‌دهد. همچنین خروجی الگوریتم ارائه شده به شکل ارقام علامت‌دار متقارن و بازه [7, -7] است اما امکان استفاده از تمام فضای کدینگ (بازه نامتقارن [7, -8]) در ورودی وجود دارد.

کلمات کلیدی: جمع دهدۀ، نمایش افزونه‌ای، رقم علامت‌دار، انتشار نقلی.

۱- مقدمه

مدارات حسابی دهدۀ با استفاده از روش‌های کدگذاری خاص براساس مدارات دودویی پیاده‌سازی می‌گردد (جدول ۱) که همین مسئله موجب پیچیدگی بیشتر مدارات دهدۀ در مقایسه با مدارات مشابه دودویی می‌شود. به طور مثال، در نمایش دهدۀ وزن هر رقم نسبت به رقم پرارزش‌تر (یا کم‌ارزش‌تر) خود ده برابر کمتر (یا بیشتر) است و همانند اعداد دودویی از توان‌های دو پیروی نمی‌کند. لذا در صورت بروز نقلی در عمل جمع، عملیات تصحیح می‌باشد صورت پذیرد که این عمل تصحیح موجب افزایش پیچیدگی می‌گردد [۱۰]. این در حالی است که عمل جمع جزء اصلی کلیه پردازش‌ها و پایه تمام عملیات حسابی است و افزایش یا کاهش سرعت در عمل جمع بر کارآیی کل سیستم تاثیرگذار است [۱۱]. به طور کلی، در نظام‌های عددی متغّر، انتشار نقلی باعث واستگی زمان جمع به طول عملوندها می‌شود که این مسئله در عملیات با دقت بالا تاخیر زیادی را تحمیل می‌کند. حال آنکه می‌توان با بهره‌گیری از نظام‌های عددی نامتغّر افزونه‌ای، عمل جمع را بدون انتشار نقلی و در زمان ثابت پیاده‌سازی نمود [۱۲]. با توجه به اهمیت عمل جمع و نقش کلیدی آن در پیاده‌سازی سایر واحدهای محاسباتی، روش‌های مختلفی برای بهبود عملکرد آن ارائه شده است. در این مقاله

پیشرفت غیرقابل تصور صنعت نیمه هادی، پیاده‌سازی سخت‌افزاری بسیاری از توابع را میسر ساخته است و در این میان به دلایل مختلف از جمله عدم نمایش دقیق برخی از اعداد (مانند ۰/۲) در نظام اعداد دودویی، همچنین به وجود آمدن نیازهای جدید در کاربردهای مالی/تجاری، موجبات ظهور مجدد سیستم‌های محاسباتی سخت‌افزاری دهدۀ را فراهم آورده و توجه بسیاری از پژوهشگران را معطوف خود ساخته است [۱]. پژوهش‌های بسیاری در خصوص مسائل و مشکلات استفاده از حساب دودویی [۱-۴]، همچنین مطالعات گسترده‌ای در مورد حجم داده‌های دهدۀ در پایگاه‌های اطلاعاتی مختلف [۵] انجام شده است، که برآیند نتایج این تحقیقات موبایل ضرورت و اهمیت به کارگیری سخت‌افزارهای حساب دهدۀ است. به دلیل اهمیت این موضوع استاندارد IEEE754، مورد بازنگری قرار گرفت و نمایش دهدۀ به عنوان بخشی از استاندارد به آن افزوده شد [۶]. همچنین پردازنده‌های تجاری گوناگونی (z900 eServer، z10 Power6، A8) با واحد اختصاصی محاسبات دهدۀ روانه بازار شده‌اند.

گردد. به این خاصیت اصطلاحاً «افزونگی ذاتی» می‌گویند. از آنجا که کدهای دارای این خاصیت از فضای کدگذاری و سخت‌افزار به شکل بهینه استفاده می‌کنند، به کارگیری آن‌ها موجب کاهش حجم سخت‌افزار و توان مصرفی مدارات مربوطه می‌گردد.

با استفاده از الگوریتم افزار مجموعه بیت‌های وزن دار [۱۳] و خاصیت افزونگی ذاتی ارقام دهدی، تقسیم بندی جدیدی مبتنی بر چهار بیت ارائه خواهد شد که با توجه به کاهش ورودی‌های توابع به کارگرفته شده، در مقایسه با الگوریتم اصلی، توان مصرفی و مساحت کمتری روزی تراشه دارد.

۱-۲- جمع دهدی افزونه‌ای

در نمایش غیرافزونه، چنانچه x و y ارقام ورودی جمع کننده و c_{in} به عنوان نقلی در نظر گرفته شود؛ حاصل جمع میانی $p = x + y + c_{in}$ محاسبه می‌شود و مقادیر حاصل جمع و نقلی خروجی c_{out} براساس مجموعه روابط (۱) محاسبه می‌شود:

$$s = |p|_{10} \text{ و } c_{out} = \begin{cases} p \\ 10 \end{cases} \quad (1)$$

به طور مشخص در این روش، برای بدست آوردن حاصل جمع هر موقعیت، مقدار نقلی ورودی از موقعیت‌های با ارزش مکانی کمتر می‌باشد مشخص گردد (انتشار نقلی). حال آن‌که با استفاده از نمایش افزونه‌ای این انتشار به طور کامل حذف می‌شود. الگوریتم کلی جمع بدون انتشار نقلی برای یک نظام عددی با ارقام علامت‌دار در بازه $[-a, \theta]$ و مبنای r به شکل زیر قابل ارائه است:

الگوریتم ۱ جمع بدون انتشار نقلی ارقام علامت‌دار

ورودی‌ها و خروجی: اعداد k رقمی با نمایش ارقام علامت‌دار در مبنای r که ورودی‌های به شکل $x_0 \dots x_{k-1}$ و $y_0 \dots y_{k-1}$ به شکل $S = s_{k-1} \dots s_0$ نمایش داده می‌شود [۱۲].

تمام رقم‌های اعداد ورودی و خروجی در بازه $[-\alpha, \theta]$ قرار دارند و سه مرحله زیر روی تمام ارقام به صورت موازی اجرا می‌شود:

۱. به دست آوردن حاصل جمع $p_0 \dots p_{k-1}$ (۰). هر رقم $p_i = x_i + y_i$.

p_i در بازه $[-2\alpha, 2\theta]$ قرار دارند.

۲. تجزیه هر p_i به حاصل جمع موقتی w_i و رقم انتقالی t_{i+1} به نحوی که شرایط $w_i \leq \theta + t_{max}$ و $p_i = w_i + r \times t_{i+1}$ برقرار باشند.

۳. به دست آوردن حاصل جمع نهایی به صورت: $s_i = w_i + t_i$.

لازم به ذکر است انتخاب بازه رقم انتقالی $[t_{min}, t_{max}]$ باید به نحوی باشد که در مرحله سوم رقم انتقالی جدیدی تولید نشود. در نظام اعداد ارقام علامت‌دار متداول $t_{min} = 1$ و $t_{max} = -1$ است. این شرایط می‌شوند. مقدار r در نمایش دهدی برابر 10 و در نمایش‌های مبنای دو برابر 2^h که $2 < h < 10$ است، انتخاب می‌شود.

۲-۲- جمع کننده‌های دهدی افزونه‌ای

در حساب دهدی، به دلیل امکان انتخاب مجموع ارقام مختلف، هم‌چنین روش‌های کدگذاری متفاوت، الگوریتم‌های متنوعی ارائه شده است که در ادامه توضیح مختصری در مورد هر یک عرضه می‌گردد. لازم به ذکر است با توجه به اینکه در الگوریتم ۱ نیاز به سه عمل جمع وجود دارد، در الگوریتم‌های زیر به نحوی سعی بر کم کردن و ساده‌سازی الگوریتم اصلی شده است.

- اولین روش جمع افزونه‌ای برای محاسبات دهدی در مقاله [۱۵] ارائه گردید. در این روش هر رقم در بازه $[6, -6]$ است و از پنج بیت برای نمایش هر رقم استفاده می‌گردد. در این شیوه کد گذاری هر مقدار با سه برابر خود و در صورت منفی بودن با سه برابر منهای 31 ، کد می‌شود و به دلیل متقاضی بودن

جدول ۱- روش‌های مختلف کدگذاری ارقام دهدی

ردیف	نام کد یا مرجع	بازه	کدینگ	α	θ	ζ	p	γ
۱	۸۴۲۱	$[0, 9]$	●●●●	۰	۹	۱۰	۰	۶
۲	افزون بر	$[3, 12]$	●●●●	۳	۱۲	۱۰	۰	۶
۳	۴۲۲۱	$[0, 9]$	●●●●	۰	۹	۱۰	۰	۰
۴	[۱۴]	$[-5, 5]$	●●●●●	-۵	۱۱	۱	۱	۲۰
۵	[۱۵]	$[-6, 6]$	●●●●●	-۶	۱۳	۳	۳	۱۸
۶	[۱۶و۱۳]	$[-7, 7]$	○●●●	-۷	۱۵	۵	۱	۱
۷	[۱۷]	$[-9, 9]$	○○○○○	-۹	۱۹	۹	۹	۳۶
۸	[۱۹و۱۸]	$[0, 15]$	●●●●	۰	۱۵	۱۶	۰	۰
۹	[۲۱و۲۰]	$[0, 10]$	●●●●	۰	۱۱	۱	۱	۶
۱۰	[۲۳و۲۲]	$[0, 18]$	●●●●	۰	۱۹	۹	۹	۰
۱۱	[۲۴]	$[-8, 9]$	○○○○	-۸	۱۸	۸	۸	۰
۱۲	[۲۵]	$[-9, 7]$	○○○○○	-۹	۱۷	۷	۷	۰

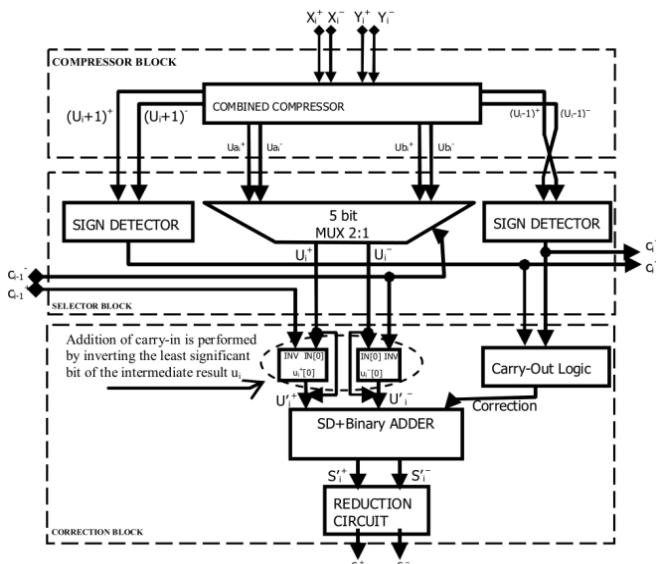
در ادامه این مقاله، در بخش دوم، کلیاتی در مورد الگوریتم جمع دهدی افزونه‌ای و هم‌چنین کدینگ‌های مختلف و جمع کننده‌های افزونه‌ای پیشین ارائه می‌شود. در بخش سوم به ارائه الگوریتم افزار مجموعه بیت‌های وزن دار خواهیم پرداخت و در بخش چهارم می‌بینیم که این روش می‌تواند از چهار بیت بازه مطرح و با چند مثال تشریح خواهد شد. در بخش پنجم، کلیه الگوریتم‌ها از جوانب مختلف مورد مقایسه و ارزیابی قرار خواهد گرفت و در پایان، در بخش ششم نتیجه‌گیری مقاله ارائه می‌شود.

۲- پیش‌زمینه

همان‌طور که در بخش پیشین بدان اشاره شد، برای نمایش اعداد دهدی با استفاده از مدارات دودوبی روش‌های کدگذاری مختلفی استفاده می‌شود. برخی از متدائل‌ترین روش‌های مورد استفاده به همراه خصوصیات منحصر به فرد هر کد، در جدول (۱) نشان داده شده است. این خصوصیات عبارتند از: کوچک‌ترین مقدار قابل نمایش (۰)، بزرگ‌ترین مقدار قابل نمایش (θ)، تعداد نمادهای مجاز (ζ)، شاخص افزونگی (p)، تعداد حالتی که مورد استفاده قرار نمی‌گیرند (γ) و نمایش نقطه‌ای هر رقم که با عنوان کدینگ در جدول ۱ مشخص شده است.

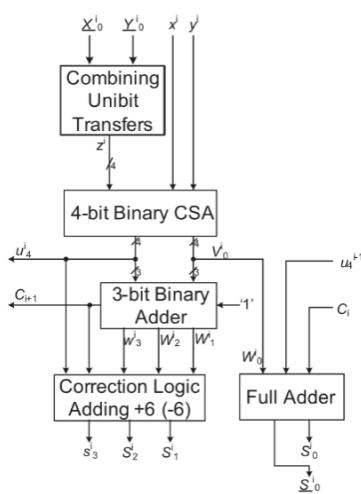
با مطالعه دقیق جدول ۱ اختلاف اساسی مبنای h با سایر مبنایهای رایج (که عموماً توانی از دو هستند) آشکار می‌شود. در واقع برای نمایش اعداد در نظام‌های افزونه‌ای نیاز به فضای ذخیره‌سازی بیشتری به منظور نمایش بیت‌های افزونه وجود دارد، اما در صورتی که مبنای مورد استفاده توافقی از دو نباشد لزوماً این قاعده بر قرار نخواهد بود. برای مثال در حساب دهدی، حداقل به چهار بیت برای نمایش یک رقم غیرافزونه‌ای که ده مقدار مختلف می‌تواند داشته باشد نیاز است (ردیفهای ۶ و ۸ در جدول ۱). لذا، شش حالت از شانزده حالت فضای کد گذاری استفاده نمی‌شود و می‌تواند به عنوان نمادهای لازم برای نمایش افزونه‌ای استفاده

[۲۶]، با استفاده از الگوریتم و چارچوب کلی ارائه شده در [۱۷]، سعی در بهبود این ایده شده است. طرح بهبود یافته در شکل ۳ نشان داده شده است.



شکل ۳- جمع کننده ددهی افزونه‌ای ارائه شده در [۲۶]

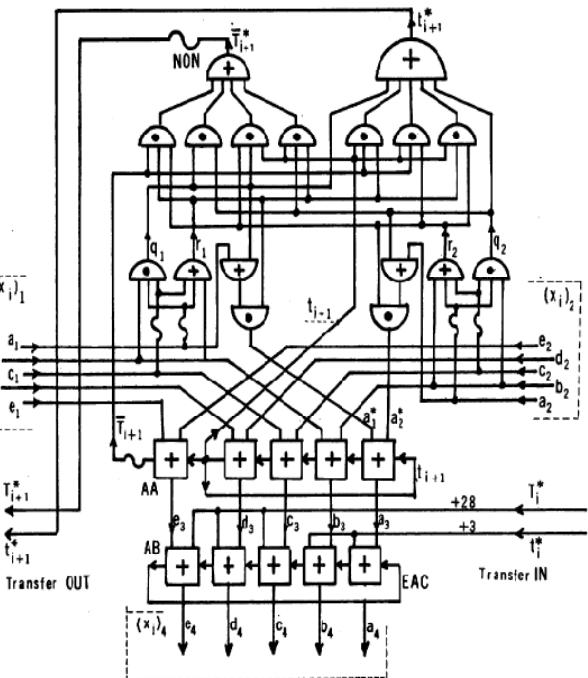
- در مقاله [۱۳] با استفاده از افزار مجموعه بیت‌های وزن دار عملیات جمع بر روی عملوند هایی در بازه [۷-۷] و نمایش مکمل دو، صورت می‌پذیرد. نتایج تحلیل و سنتز صورت گرفته نشان دهنده برتری این الگوریتم در تمامی پارامترهای طراحی است. این الگوریتم در بخش سوم به طور مفصل مورد تجزیه و تحلیل قرار می‌گیرد.
- در [۲۴]، با به کارگیری نمایش یونی بیت و استفاده از مجموع ارقامی در بازه [۸-۹]، در واقع با یک کدگذاری پنج بیتی، الگوریتمی کارا برای جمع ددهی افزونه ارائه شده است (شکل ۴).



شکل ۴- جمع کننده ددهی افزونه‌ای ارائه شده در [۲۴]

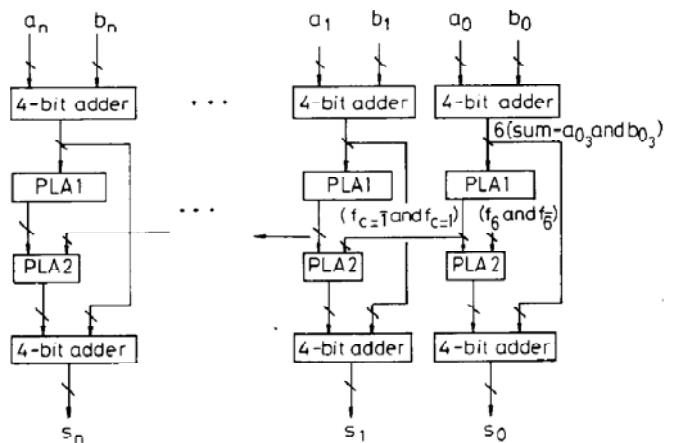
- در مقاله [۲۵] با استفاده از روش ارائه شده در [۱۳] و ارقامی در بازه [۹-۹]، با استفاده از پنج بیت الگوریتم جدیدی برای جمع افزونه‌ای ددهی ارائه شده است. به دلیل استفاده از پنج بیت به منظور کدینگ ارقام و توابع بزرگ‌تر در مسیر بحرانی تاخیر این جمع کننده (F2)، تقریباً در تمامی مقایسه‌ها، عملکردی به مراتب کمتر از [۱۳] دارد. شما از این جمع کننده در شکل ۵ نشان داده شده است.

بازه و خود مکمل بودن کد، عمل منفی کردن به آسانی قابل انجام است. شما از این جمع کننده در شکل ۱ نشان داده شده است.



شکل ۱- جمع کننده ددهی افزونه‌ای ارائه شده در [۱۵]

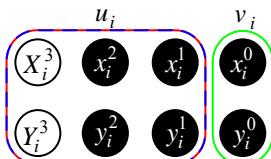
- در مقاله [۱۶]، با استفاده از کدگذاری مکمل دو، از ارقامی در بازه [۷-۷] برای نمایش اعداد ددهی افزونه‌ای (RBCD)^۱ استفاده شده است. این روش نیز کاملاً مبتنی بر الگوریتم کلی جمع ارقام علامت دار بوده و از دو PLA برای پیاده‌سازی دوتابع تولید کننده رقم انتقالی و مقدار تصحیح استفاده می‌کند (شکل ۲).



شکل ۲- جمع کننده ددهی افزونه‌ای ارائه شده در [۱۶]

- در مقاله [۱۷]، با استفاده از نمایش اعداد در بازه متقاضان [۹-۹] یک جمع کننده جدید افزونه‌ای ارائه شده است. در این روش هر رقم از ترکیب دو رقم BCD یکی با وزن مثبت و دیگری با وزن منفی تشکیل شده است و ارزش هر رقم ددهی با کم کردن قسمت منفی از قسمت مثبت به دست می‌آید $Z_i = Z_i^+ - Z_i^-$. با وجود اینکه این نظام از نظر سخت‌افزاری، سربار بسیار زیادی دارد اما در مواردی همانند منفی کردن و در نتیجه آن تفریق کردن، همچنین در تبدیلات، دارای کارآیی مطلوبی است. همچنین در مقاله

با این روش افزار مقتضی دار $\|v_i\| \in [0, 2]$ و رقم انتقالی دهدۀ $t_{i+1} \in \{-1, 0, 1\}$ استخراج و مقدار باقیمانده با توجه به رابطه $\|z_i\| = \|u_i\| - 10 \times t_{i+1}$ می‌گردد. در مرحله بعد مقدار حاصل جمع میانی $\|v_i\| + \|v_i\|$ با مقدار $w_i = \|z_i\| + \|v_i\|$ با استفاده از مقادیر v_i و z_i به دست می‌آید. این مقدار در بازه $[-6, 6] = [-6, 4] + [0, 2]$ قرار دارد که در نهایت با اضافه شدن مقدار رقم انتقالی موقعیت کم ارزش‌تر، حاصل جمع نهایی $s_i \in [-7, 7]$ حاصل می‌گردد.

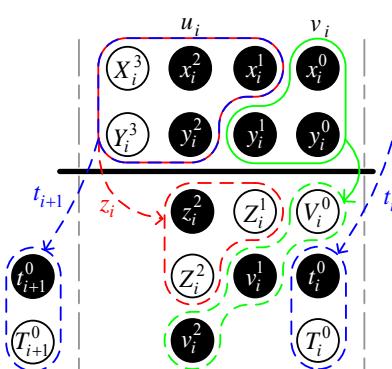


شکل ۷- افزار p_i با استفاده از شش بیت

در افزار مبتنی بر شش بیت، تعداد ورودی‌های تابع تولید کننده رقم انتقالی و همچنین مقدار باقیمانده از هشت به شش کاهش می‌یابد که این امر علاوه بر افزایش سرعت، موجب کاهش مساحت بر تراشه و توان مصرفی نیز می‌گردد. در بخش بعد سعی بر کاهش بیشتر تعداد ورودی‌های تابع فوق الذکر خواهد بود.

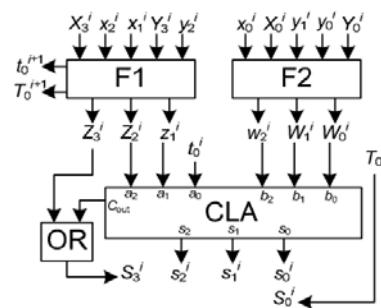
۲-۳- افزار مبتنی بر پنج بیت

چنانچه افزار مجموعه بیت‌های p_i همانند شکل ۸ صورت پذیرد، تعداد ورودی‌های تابع تولید کننده رقم انتقالی و مقدار باقیمانده پنج بیت خواهد بود. در شکل ۸ جزئیات بیشتری از نحوه استخراج z_i و t_{i+1} به همراه کدینگ مورد استفاده نشان داده است.



شکل ۸- افزار p_i و استخراج t_{i+1} و z_i با کدینگ

با افزار پنج بیتی، مقدار v_i افزایش می‌یابد ($\|v_i\| \in [0, 4]$) و مقدار دوازده از بازه u_i خارج می‌شود. در این صورت، $\{ -1, 0, 1 \} \in \|z_i\| \in t_{i+1} \in \{-6, -4, -2, 0, 2\}$ خواهد شد که همانند افزار شش بیتی حاصل جمع میانی در بازه $[-6, 6] = [-6, 4] + [0, 2]$ به دست می‌آید و در نهایت حاصل جمع نهایی در بازه مطلوب است، بوسیله دو مدار ترکیبی مقدار v_i به $v_i^2 v_i^1 v_i^0$ تبدیل و مقادیر $(Z_i^1 Z_i^2 z_i^2) z_i$ از u_i به دست می‌آیند. نحوه کدینگ به گونه‌ای در نظر گرفته شده است که علاوه بر حفظ درستی مقدار نمایش داده شده، مقدار



شکل ۵- جمع کننده دهدۀ افزونهای ارائه شده در [۲۵]

۳- الگوریتم افزار بیت‌های وزن دار

در الگوریتم جمع افزونهای با استفاده از افزار بیت‌های وزن دار، مرحله اول الگوریتم ۱، بدون هزینه و با استفاده از نمایش ذخیره نقلی انجام می‌شود. این حاصل جمع میانی با استفاده از مجموعه ارقام $[7, -7]$ در شکل ۱ نشان داده شده است. این مجموعه، «ارقام غلامتدار دهدۀ هفت‌تایی»^۳ و به اختصار DSSD نام‌گذاری شده و مشابه روش کدگذاری $[16, 7, 0]$ هر رقم به صورت مکمل دو نمایش داده می‌شود. لازم به ذکر است که این نحوه نمایش دارای افونگی ذاتی است و علاوه بر جمع، امکان تفريع را نیز به آسانی فراهم می‌نماید. در شکل ۶ بیت‌های غلامت، که بیت‌هایی با وزن منفی هستند و اصطلاحاً به آن‌ها نگایبت^۴ گفته می‌شود به شکل دوایر توخالی (حروف بزرگ) و سایر بیت‌ها که و اصطلاحاً به آن‌ها پوزی بیت^۵ می‌شود با دوایر توپر (حروف کوچک) نشان داده شده است.



شکل ۶- نمایش p_i به شکل ذخیره‌نقلی مکمل دو

در مرحله دوم الگوریتم ۱، از مقایسه حاصل جمع میانی با بازه مجاز ارقام، حاصل جمع موقتی w_i و رقم انتقالی t_{i+1} با در نظر گرفتن شرایط مربوطه استخراج می‌شود. با توجه به اینکه عمل مقایسه به نوعی پیچیدگی عمل جمع را دارد و موجب افزایش زمان محاسبه می‌شود، راهکارهای مختلفی برای اجتناب از مقایسه به کارگرفته می‌شود که در ادامه روش افزار، به عنوان روش ارائه شده در [۱۲] و بهبود یافته در این مقاله، مورد مطالعه قرار گرفته است.

۳-۱- افزار مبتنی بر شش بیت

از نظر تئوری امکان استخراج مقدار رقم انتقالی t_{i+1} با استفاده مجموعه بیت‌های p_i وجود دارد، اما در صورت استفاده از تمام بیت‌های این مجموعه، با یک تابع منطقی هشت ورودی روبرو خواهیم بود که در عمل از نظر پیاده‌سازی شرایط مطلوبی را فراهم نخواهد ساخت. لذا با افزار مجموعه بیت‌های p_i سعی در کم کردن تعداد ورودی‌های تابع فوق الذکر خواهیم نمود. در شکل ۷ افزار مجموعه بیت‌های p_i به دو مجموعه u_i و v_i نشان داده شده است؛ $\|u_i\| = \|v_i\| = 4$. بهطوری‌که u_i شامل سه بیت پرازش عملوندها و v_i شامل دو بیت باقیمانده است.

با مقایسه شکل ۱۰ و ۱۱، سادگی بخش جمع کننده در شکل ۱۱ به دلیل استفاده از کدینگ $Z_i^3 Z_i^2 Z_i^1$ قابل مشاهد است (یک نیم جمع کننده کمتر استفاده شده است). لازم به ذکر است در پیاده‌سازی واقعی هر دو شکل از جمع کننده پیش‌بینی نقلی استفاده می‌شود.

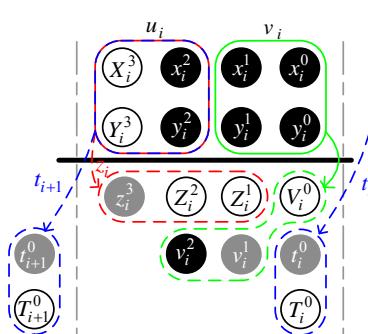
از آن جایی که عملوندها در نمایش DSSD به شکل مکمل دو کد می‌شوند، به آسانی با معکوس کردن بیت‌های مفروق و افزودن مقدار یک به شکل نقلی ورودی عمل تفیری قابل انجام است. لازم به ذکر است که برای راحتی کار و عدم نیاز به گیت‌های معکوس کننده، نگابیت‌ها به شکل معکوس کدگذاری شده‌اند (مقدار صفر منطقی نشان دهنده ارزش حسابی منفی یک و مقدار یک منطقی نشان دهنده مقدار حسابی صفر است) [۲۷].

۴- الگوریتم افزار مجموعه بیت‌های وزن دار با تقسیم‌بندی چهار بیتی

در بخش ۲-۳ افزار پنج بیتی مجموعه بیت‌های p_i نشان داده شد. چنانچه عمل افزار به شکل چهار بیتی انجام شود، مقدار v_i در بازه $[0, 6]$ [۶] قرار می‌گیرد و مقدار u_i به بازه $\{8, -16, -12, -8, -4, 0, 4, 8\}$ محدود می‌شود. در این افزار، اگر $\|u_i\| = 8$ باشد، برای استخراج مقدار صحیح رقم انتقالی، مقدار v_i تعیین کننده خواهد بود. لذا همان طور که در [۱۹] نشان داده شده است، از نظر تثویری امکان افزار با چهار بیت وجود ندارد مگر اینکه حالات خاص ایجاد شده به نوعی با کمترین تاثیر بر کارآیی مدیریت شود.

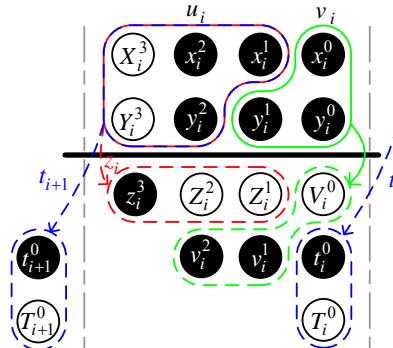
اگر به‌طور پیش فرض در حالت $-8 = \|u_i\|$ مقدار $t_{i+1} = -1$ و $2 = \|z_i\|$ در نظر گرفته شود، چنانچه $4 > \|v_i\|$ باشد، با در نظر گرفتن رقم انتقالی ورودی امکان تولید مقداری خارج از بازه $[7, -7]$ وجود خواهد داشت. از آنجا که تنها حالت مشکل ساز، در $-8 = \|u_i\| > 4 = \|v_i\|$ حادث می‌شود، این حالت به عنوان یک استثنای شناسایی شده و اقدام لازم برای رفع آن صورت می‌پذیرد.

به طور کلی، در حالت $-8 = \|u_i\|$ مقدار $-1 = t_{i+1}$ با یک نگابیت با کدگذاری معکوس $T_{i+1}^0 = 0$ و یک پوزی بیت $= 0 = t_{i+1}^0$ ، هم‌چنین $2 = \|z_i\|$ با $Z_i^3 = 1$ و $Z_i^2 = 0$ ، $Z_i^1 = 0$ باشد، در واقع حالت استثنای خارج از بازه $\|v_i\| > 4$ مقدار $V_i^1 = 0$ باشد، در صورت بروز استثنای همواره یک است می‌بایست مقدار t_{i+1} و $t_{i+1}^0 = 0$ و $t_{i+1}^0 = 1$ تصحیح گردد. برای انجام تصحیح رقم انتقالی، کافیست مقدار $Z_i^3 = -8$ شود. اما برای تصحیح مقدار z_i علاوه بر صفر کردن مقدار Z_i^3 ، مقدار $V_i^1 = 0$ که در صورت بروز استثنای همواره یک است می‌بایست صفر گردد. بیت‌های تحت تأثیر استثنای در شکل ۱۲ با رنگ خاکستری نشان داده شده است.



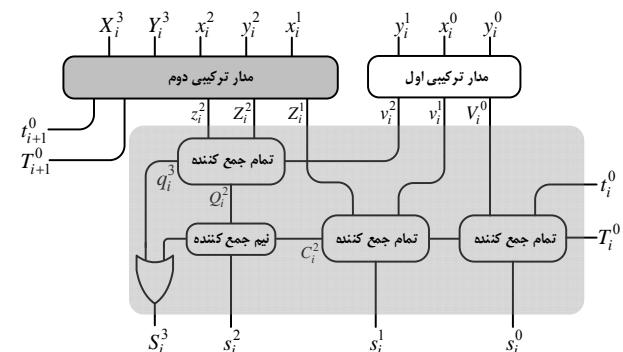
شکل ۱۲- افزار چهار بیتی p_i و استخراج z_i و t_{i+1}

خروجی نیز در بازه و کدینگ مطلوب حاصل شود ($S_i^0 S_i^1 S_i^2 S_i^3$). به طور مثال به جای شکل ۸ می‌توان از شکل ۹ نیز استفاده نمود. در این افزار نحوه کدینگ مقدار v_i به $t_{i+1}^0 T_{i+1}^0$ تبدیل و مقدار t_{i+1} قبل از u_i می‌باشد و تفاوت در کدینگ z_i می‌باشد. زیرا در این کدینگ به صورت $Z_i^3 Z_i^2 Z_i^1$ کدگذاری می‌کنیم و در این حالت نیز نحوه کدینگ به گونه‌ای در نظر گرفته شده است که علاوه بر حفظ درستی مقدار نمایش داده شده، مقدار خروجی نیز در بازه و کدینگ مطلوب حاصل شود ($S_i^0 S_i^1 S_i^2 S_i^3$).



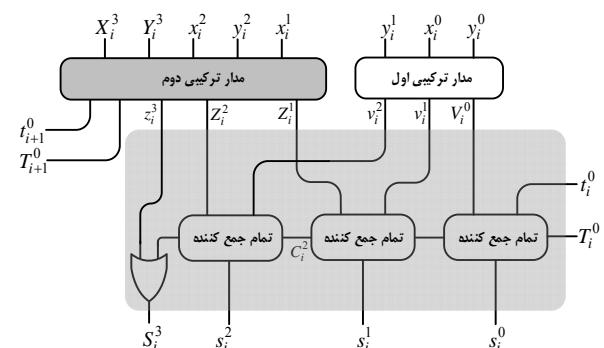
شکل ۹- افزار p_i و استخراج z_i و t_{i+1} با کدینگ

چنانچه z_i مطابق با شکل ۸ استخراج گردد، شمای کلی مدار جمع دهدھی دو رقم در بازه $[7, 7]$ [۷-۷]، همانند شکل ۱۰ خواهد بود.



شکل ۱۰- مدار جمع کننده دو رقم در بازه $[7, 7]$ [۷-۷] با کدینگ

حال اگر z_i مطابق با شکل ۹ استخراج گردد، شمای کلی مدار جمع دهدھی دو رقم در بازه $[7, 7]$ [۷-۷]، همانند شکل ۱۱ می‌گردد.

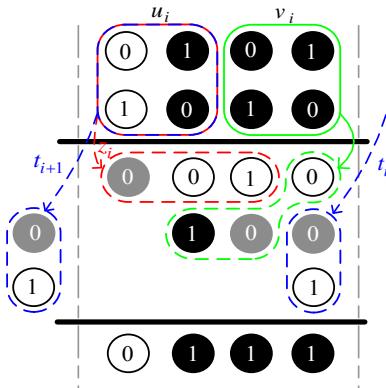


شکل ۱۱- مدار جمع کننده دو رقم در بازه $[7, 7]$ [۷-۷] با کدینگ

می‌باید. لازم به ذکر است برای پیاده سازی تفیریق فقط تا خیر گیت XOR به مسیر بحرانی تا خیر افزوده می‌شود.
برای روش‌تر شدن افزار چهار بیتی دو مثال در حالت‌های عدم رخ داد استثنا و رخ داد استثنا در ادامه مورد بررسی قرار می‌گیرد.

مثال ۱ (جمع دو عدد افزونه‌ای با استفاده از افزار چهار بیتی- حالت عدم رخ داد استثنا)

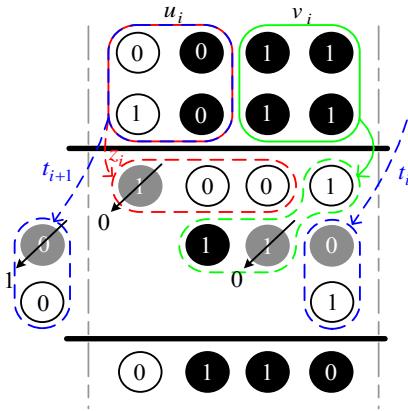
فرآیند جمع دو عدد $\bar{3}$ و 2 در شکل ۱۳ نشان داده شده است. دو عدد بر اساس افزار چهار بیتی، تقسیم بندی شده و پس از محاسبه بیت‌های $z_i^3 z_i^2 z_i^1$ و $v_i^2 v_i^1 v_i^0$ و با توجه به عدم رخ داد استثنا جمع نهایی صورت می‌گیرد. در این مثال رقم نقلی ورودی صفر در نظر گرفته شده است. به دست می‌آید ($\bar{3} + 2 = \bar{1}$).



شکل ۱۳- عدم رخ داد استثنا در افزار چهار بیتی

مثال ۲ (جمع دو عدد افزونه‌ای با استفاده از افزار چهار بیتی- حالت رخ داد استثنا)

در این مثال، دو عدد $\bar{5}$ و 3 برای جمع انتخاب شده است. برخلاف مثال قبل برای جمع این دو عدد حالت استثنا رخ می‌دهد و قبل از جمع نهایی، همان‌طور که در شکل ۱۴ نشان داده شده است تصحیح صورت می‌پذیرد.



شکل ۱۴- رخ داد استثنا در افزار چهار بیتی

لازم به ذکر است در محاسبه بیت‌های t_{i+1}^0 ، t_i^3 و $v_i^2 v_i^1 v_i^0$ ، حالت استثنا لحاظ می‌گردد و این بیت‌ها به شکل صحیح بدست می‌آید، در شکل ۱۴ برای درک بهتر مقدار قبل و بعد از تصحیح نشان داده شده است.

در ادامه جزئیات فرآیند فوق الذکر همراه با روابط منطقی لازم برای پیاده‌سازی مدار مربوطه در روابط (۲) تا (۶) آمده است. در واقع بعد از تشکیل p_i و افزار آن به دو مجموعه u_i و v_i ، چهار مرحله زیر برای رسیدن به حاصل جمع نهایی متصور است. قسمت‌های مختلف این مراحل دارای همپوشانی زمانی هستند و معادلات لازم برای پیاده‌سازی توابع منطقی در هر مرحله ذکر شده است. تمام روابط با استفاده از جدول درستی به راحتی قابل استخراج هستند.

- برای تشخیص حالت استثنا، در چهار بیت کم ارزش رخ دادن مقدار پنج یا شش ($\|v_i\| > 4$) بررسی می‌شود (E_i^1) و در چهار بیت پارازش منفی هشت ($\|u_i\| = -8$) بررسی می‌گردد (E_i^h). در صورت مثبت بودن هر دو، استثنا رخ داده است ($E_i = E_i^1 E_i^h$).

$$\begin{aligned} E_i^1 &= x_i^1 y_i^1 (x_i^0 \vee y_i^0) \\ (E_i^h) &= (X_i^3 \oplus Y_i^3) \overline{x_i^2 y_i^2} \vee \overline{X_i^3 Y_i^3} \overline{x_i^2 y_i^2} \end{aligned} \quad (2)$$

- استخراج t_{i+1} و z_i از u_i با توجه به رخ داد یا عدم رخ داد استثنا و در نظر گرفتن $\|t_{i+1}\| + \|z_i\| = 10$. لازم به ذکر است که مقدار بیت‌های z_i^3 ، Z_i^2 ، T_i^1 و T_{i+1}^0 مستقل از استثنا محاسبه می‌شوند و فقط بیت‌های t_{i+1}^0 و v_i^1 از رخ داد استثنا تاثیر می‌پذیرند (در شکل ۱۲ نشان داده شده است).

$$\begin{aligned} t_{i+1}^0 &= (X_i^3 \vee Y_i^3)(x_i^2 \vee y_i^2) \vee X_i^3 Y_i^3 \vee \\ &\quad E_i^1 (X_i^3 \vee Y_i^3 \vee x_i^2 y_i^2) \\ T_{i+1}^0 &= X_i^3 Y_i^3 (x_i^2 \vee y_i^2) \end{aligned} \quad (3)$$

$$\begin{aligned} z_i^3 &= E_i^h \overline{E_i^1} \\ Z_i^2 &= (X_i^3 \vee Y_i^3) x_i^2 y_i^2 \vee (x_i^2 \oplus y_i^2) \overline{X_i^3 Y_i^3} \vee \\ &\quad X_i^3 Y_i^3 \overline{x_i^2 y_i^2} \\ Z_i^1 &= (X_i^3 \oplus Y_i^3) (x_i^2 \vee y_i^2) \vee X_i^3 Y_i^3 \overline{x_i^2 y_i^2} \end{aligned} \quad (4)$$

- تبدیل v_i به $v_i^2 v_i^1 v_i^0$ ، با در نظر گرفتن حالت استثنا برای v_i^1 .

$$\begin{aligned} V_i^0 &= \overline{x_i^0 \oplus y_i^0} \\ V_i^1 &= \overline{E_i} (y_i^1 \oplus (x_i^0 \vee y_i^0)) \\ V_i^2 &= (x_i^0 \vee y_i^0) (x_i^1 \vee y_i^1) \vee x_i^1 y_i^1 \end{aligned} \quad (5)$$

- تمام مقدابر تولید شده به وسیله یک جمع کننده سه بیتی جمع شده و بیت‌های s_i^0 ، s_i^1 و s_i^2 را تولید می‌کنند و نقلی خروجی جمع کننده به وسیله «یای» منطقی با z_i^3 بیت S_i^3 را تولید می‌نماید. در روابط نشان داده شده مقادیر نقلی به شکل ساده و با فرض استفاده از جمع کننده انتشار نقلی موجی نشان داده شده است که در عمل، از جمع کننده پیش‌بینی نقلی استفاده می‌شود.

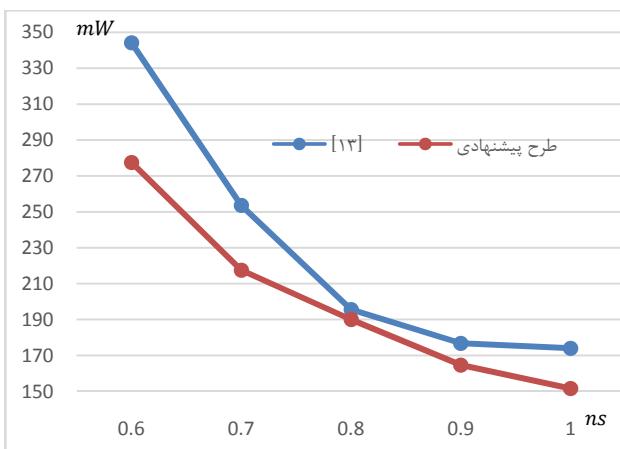
$$\begin{aligned} S_i^0 &= (V_i^0 \oplus t_i^0) \oplus T_i^0 C_i^1 = \text{carry } (V_i^0, T_i^0, t_i^0) \\ S_i^1 &= (V_i^1 \oplus Z_i^1) \oplus C_i^1 C_i^2 = \text{carry } (Z_i^1, C_i^1, V_i^1) \\ S_i^2 &= (V_i^2 \oplus Z_i^2) \oplus C_i^2 C_i^3 = \text{carry } (Z_i^2, C_i^2, V_i^2) \\ S_i^3 &= z_i^3 \vee C_i^3 \end{aligned} \quad (6)$$

برای یک‌پارچه ساختن واحد جمع کننده و تغیریق کننده می‌توان از یک سیگنال کنترلی مانند Sub استفاده نمود. با استفاده از این سیگنال کنترلی در صورت نیاز به انجام عمل تغیریق بیت‌های مفروق به وسیله گیت XOR معکوس می‌شود و با در نظر گرفتن سیگنال Sub در روابط (۵) مقدار $V_i^2 V_i^1 V_i^0$ یک واحد افزایش

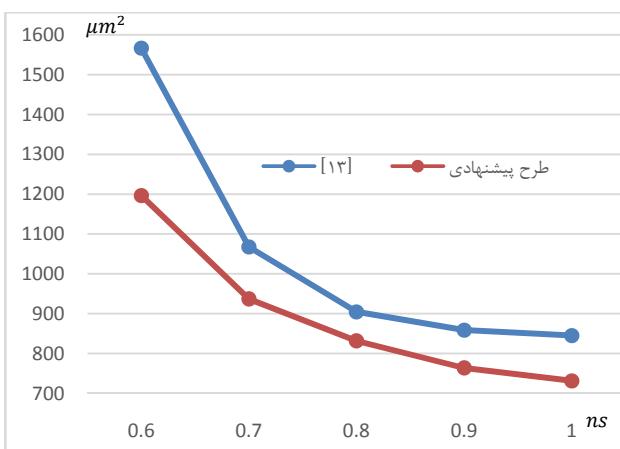
مانند ضرب کننده‌ها هستند، حتی بهبودهای کوچک نیز تأثیر بهسزایی در کارآیی کل واحد محاسباتی دارد.

جدول ۳- نتایج سنتز جمع‌کننده‌های افزونهای دهدی

مساحت (μm^2)	توان مصرفی ایستا (pW)	توان مصرفی پویا (mW)	تاخیر (ns)	مرجع
1100	37.24	1.17	0.69	[۱۳]
1901	70.98	2.13	1.16	[۱۵]
1480	44.34	1.65	0.87	[۱۶]
3139	95.79	4.04	1.10	[۱۷]
1133	32.31	1.22	0.63	[۱۹]
1354	37.87	1.48	0.75	[۲۰]
1390	45.78	1.54	0.77	[۲۲]
1289	46.77	1.52	0.77	[۲۴]
1310	40.19	3.16	0.70	[۲۵]
1095	49.20	1.09	0.63	طرح پیشنهادی



شکل ۱۵- مقایسه توان مصرفی



شکل ۱۶- مقایسه مساحت

۵- مقایسه و ارزیابی

در این بخش به ارزیابی الگوریتم پیشنهادی و مقایسه آن با الگوریتم‌های ارائه شده برای جمع دهدی افزونهای خواهیم پرداخت. با یک بررسی ساده و استفاده از تعداد گیت‌های تشکیل دهنده مسیر بحرانی تاخیر (با در نظر گرفتن پیاده‌سازی با گیت‌های دو ورودی Δg)، مشخص می‌شود که تاخیر این مدار Δg ۱۰ است در حالی که بهترین طرح ارائه شده قبلی دارای تاخیر Δg ۱۲ است. به منظور ارزیابی دقیق‌تر، در جدول ۲ کلیه روش‌های جمع افزونهای دهدی، شامل جمع‌کننده‌های افزونهای دهدی با ارقام علامت‌دار هم‌چنین جمع‌کننده‌هایی که در کاهش حاصل ضرب‌های جزئی در ضرب‌کننده‌های دهدی مورد استفاده قرار گرفته‌اند (در واقع جمع‌کننده‌های افزونهای با ارقام بدون علامت)، با استفاده از تلاش منطقی بررسی شده است. در این جدول برای طرح‌هایی که فاصله تاخیر آن‌ها نزدیک بهم است، این نتایج با در نظر گرفتن دو فرض حداقل انشعاب و انشعاب بدون محدودیت گزارش شده است.

جدول ۲- مقایسه تاخیر و مساحت جمع‌کننده‌های افزونهای دهدی

نسبت مساحت	مساحت (NAND2)	نسبت تاخیر	تاخیر (FO4)	مرجع
1.82	184	1.13	8.56-9.96	[۱۳]
1.4	142	2.14	16.11	[۱۵]
1.21	122	1.8	13.56	[۱۶]
4.24	428	1.95	14.74	[۱۷]
1.02	103	1.04	7.86-9.20	[۱۹]
1.08	109	1.18	8.89-10.45	[۲۰]
1.92	194	1.19	9.00-12.17	[۲۳]
1.48	150	1.2	9.06-10.44	[۲۴]
1.84	186	1.18	8.87-9.98	[۲۵]
1	101	1	7.54-9.05	طرح پیشنهادی

همان‌طور که مشاهده می‌شود طرح پیشنهادی در مقایسه با جمع‌کننده‌های ارقام علامت‌دار ([۱۳]، [۱۷-۱۵] و [۲۵، ۲۴]، در حدود ۱۳ درصد از سریع‌ترین جمع‌کننده تاخیر کمتری نشان می‌دهد. هم‌چنین این جمع‌کننده کمترین مقدار مساحت را دارد (101 NAND2).

در جدول ۳، طرح‌های ذکر شده در جدول ۲، با استفاده از نتایج سنتز مورد بررسی مجدد قرار گرفته و مقایسه کاملی از تاخیر، مساحت و توان مصرفی ارائه شده است. برای سنتز طرح پیشنهادی استاندارد ۱۳۰ نانومتر TSMC استفاده شده است و شرایط برای تمامی طرح‌های یکسان در نظر گرفته شده است.

نتایج سنتز بدست آمده، موید تحلیل‌های اولیه و نتایج جدول ۲ است. در شکل‌های ۱۵ و ۱۶ نتایج سنتز طرح پیشنهادی و طرح [۱۳] برای توان مصرفی و مساحت در فاصله زمانی بین $0/6$ تا 1 نانو ثانیه نشان داده شده است. طرح پیشنهادی بدون افزایش تاخیر، مساحت و توان مصرفی کمتری دارد. با توجه به اینکه جمع‌کننده‌های افزونهای اجزای تشکیل دهنده واحدهای محاسباتی بزرگ‌تر

۶- نتیجه‌گیری

Representations," *IEEE Transactions on Computers*, vol. 39, pp. 89–98, Jan. 1990.

[13] S. Gorgin, and G. Jaberipur, "Fully Redundant Decimal Arithmetic," *Proceedings of the 19th IEEE Symposium on Computer Arithmetic, Portland, USA*, pp. 145–152, Jun. 2009.

[14] M. A. Erle, E. M. Schwartz, and M. J. Schulte, "Decimal Multiplication with Efficient Partial Product Generation," *17th IEEE Symposium on Computer Arithmetic*, pp. 21-28, Jun. 2005.

[15] A. Svoboda, "Decimal Adder with Signed Digit Arithmetic," *IEEE Transactions on Computers*, vol. C-18, no. 3, pp. 212-215, Mar. 1969.

[16] B. Shirazi, D. Y. Yun, and C. N. Zhang, "RBCD: Redundant Binary Coded Decimal Adder," *IEE Proceedings Computer & Digital Techniques (CDT)*, vol. 36, no. 2, Mar. 1989.

[17] H. Nikmehr, B. J. Phillips, and C. C. Lim, "A Decimal Carry-free Adder," *Proc. SPIE Conf. Smart Mater., Nano-, Micro-Smart Syst.*, pp. 786–797, Dec. 2004.

[18] R. D. Kenney, M. J. Schulte, and M. A. Erle, "A High-frequency Decimal Multiplier," *IEEE Int. Conf. on computer Design: VLSI in Computers and Processors (ICCD)*, pp. 26-29, Oct. 2004.

[19] S. Gorgin, and G. Jaberipur, "A Fully Redundant Decimal Adder and Its Application in Parallel Decimal Multipliers," *Microelectronics Journal*, vol. 40, Issue 10, pp. 1471-1481, Oct. 2009.

[20] M. A. Erle, and M. J. Schulte, "Decimal Multiplication via Carry-save Addition," *Conference on Application-Specific Systems, Architectures, and Processors*, pp. 348-358, Jun. 2003.

[21] T. Lang, and A. Nannarelli, "A Radix-10 Combinational Multiplier," *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, pp. 313-317, Nov. 2006.

[22] A. Vazquez, E. Antelo, and P. Montuschi, "A New Family of High-Performance Parallel Decimal Multipliers," *Proc. 18th IEEE Symposium on Computer Arithmetic*, pp. 195-204, Jun. 2007.

[23] I. D. Castellanos, and J. E. Stine, "Compressor Trees for Decimal Partial Product Reduction," *ACM Great Lakes Symposium on VLSI*, pp. 107-110, May 2008.

[24] A. Kaivani, and G. Jaberipur, "Fully Redundant Decimal Addition and Subtraction Using Stored-unibit encoding," *Integration the VLSI journal*, 2009.

[25] A. Kaivani, and S-B. Ko, "Decimal Signed Digit Addition Using Stored Transfer encoding," *26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1-4, May 2013.

در این مقاله، با توجه به اهمیت و نقش کلیدی عمل جمع، با به کارگیری نظام عددی نامتعارف افزونه‌ای، یک الگوریتم جمع افزونه‌ای دهدی مبتنی بر افزار بیت‌های وزن دار ارائه گردید. در طرح پیشنهادی از خاصیت افزونگی ذاتی موجود در ارقام دهدی استفاده شده و ارقام ورودی به شکل ارقام علامت‌دار متقارن و بازه [7-7] است. با توجه به افزار چهار بیتی صورت گرفته، ورودی تابع داخلی جمع کننده به چهار محدود شده و امکان پیاده‌سازی با تعداد گیت‌های کمتر فراهم گردید. برای مقایسه، بررسی جامعی روی تمامی جمع کننده‌های افزونه‌ای دهدی صورت پذیرفت و به دو شکل تحلیلی و مبتنی بر سنتز، طرح پیشنهادی مورد بررسی قرار گرفت. نتایج نشان دهنده کاهش توان مصرفی و مساحت بر تراشه بدون از دست دادن کارآیی می‌باشد.

مراجع

- [1] M. F. Cowlishaw, and et. al., "Decimal Floating-Point Specification," *Proc. 15th IEEE Symposium on Computer Arithmetic*, pp. 147-154, Jun. 2001.
- [2] IBM Corporation, "The 'telco' Benchmark," <http://www2.hursley.ibm.com/decimal/telcoSpec.html>, Mar. 2003.
- [3] K. Quinn, "Ever Had Problems Rounding off Figures? This Stock Exchange Has," *Wall Street Journal*, Nov. 1983.
- [4] M. F. Cowlishaw, "Decimal Arithmetic FAQ," <http://speleotrove.com/decimal/decifaq>.
- [5] A. Tsang, and M. Olschanowsky, "A Study of Database 2 Customer Queries," *IBM Technical report 03.413*, IBM, San Jose, CA, Apr. 1991.
- [6] Institute of Electrical and Electronics Engineers, *IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008*, Aug. 2008.
- [7] F. Y. Busaba, and et. al., "The IBM z900 Decimal Arithmetic Unit," *Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1335-1339, Nov. 2001.
- [8] S. Shankland, "IBM's POWER6 Gets Help with Math, Multimedia," *ZDNet News*, Oct. 2006.
- [9] C. F. Webb, "IBM z10: The Next-Generation Mainframe Microprocessor," *IEEE Micro*, vol. 28, Issue 2, pp. 19-29, 2008.
- [10] R. K. Richards, "Arithmetic Operations in Digital Computers," Van Nostrand, New York, 1955.
- [11] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford, 2nd ed., 2010.
- [12] B. Parhami, "Generalized Signed-Digit Number Systems: A Unifying Framework for Redundant Number

[26] J. Moskal, E. Oruklu, and J. Saniie, "Design and Synthesis of a Carry-Free Signed-Digit Decimal Adder," *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS'07)*, pp. 1089-1092, May 2007.

[27] G. Jaberipur, and B. Parhami, "Posibits, Negabits, and Their Mixed Use in Efficient Realization of Arithmetic Algorithms," *15th Symposium on Computer Architecture and Digital Systems*, pp. 3-9, Sep. 2010.

سعید گرگین مدرک کارشناسی و کارشناسی ارشد خود را در سال های ۱۳۸۱ و ۱۳۸۴ از دانشگاه آزاد اسلامی اخذ نمود و در سال ۱۳۸۹ موفق به اخذ مدرک دکتری از دانشگاه شهید بهشتی گردید. در حال حاضر او استادیار پژوهشکده برق و فناوری اطلاعات سازمان پژوهش های عملی و صنعتی ایران است. همچنین به عنوان محقق در پژوهشکده کامپیوتر پژوهشگاه دانش های بنیادی مشغول به فعالیت می باشد. در کنار موضوعات تحقیقات کاربردی در حوزه فناوری اطلاعات، سایر زمینه های مورد علاقه او عبارتند از: حساب کامپیوترا، طراحی VLSI، سیستم های پردازش سریع و پردازش موازی.



آدرس پست الکترونیکی ایشان عبارت است از:
gorgin@irost.org

لیلی میرمقتدایی مدرک کارشناسی خود را از دانشگاه آزاد اسلامی در سال ۱۳۸۷ و کارشناسی ارشد خود را از دانشگاه شهید بهشتی در سال ۱۳۹۵ اخذ نموده است. تحقیقات کارشناسی ارشد او در حوزه حساب کامپیوترا متتمرکز بوده و موضوع پایان نامه او طراحی و پیاده سازی جمع کننده های ممیز شناور افزونه های دهدھی است. سایر موضوعات مورد علاقه او ساخت افزارهای با قابلیت پیکربندی مجدد، سیستم های نهفته و تراشه های زیستی است.



آدرس پست الکترونیکی ایشان عبارت است از:
leileemirmoghtadaei@gmail.com

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۵/۰۳/۰۳

تاریخ اصلاح: ۱۳۹۵/۰۵/۱۰

تاریخ قبول شدن: ۱۳۹۵/۰۶/۲۳

نویسنده مرتبط: دکتر سعید گرگین، پژوهشکده برق و کامپیوتر، سازمان پژوهش های علمی و صنعتی ایران، تهران، ایران.

¹Redundant BCD

²Decimal Septa Signed Digit (DSSD)

³Negabit

⁴Posibit

طراحی و مشخصه‌یابی بلوک منطقی قابل پیکربندی تمام گرافنی قابل استفاده در تراسه‌های FPGA

حسین کریمیان علی‌داش

سیدعلی سیف کاشانی

دانشکده برق و کامپیوتر، دانشگاه کاشان، کاشان، ایران

چکیده

یکی از لوازم حیاتی برای دست‌یابی به پیشرفت تکنولوژی در صنایع میکروالکترونیک، کوچکتر و سریعتر شدن افزارهای الکترونیکی است. از طرفی مطابق نقشه راه بین‌المللی صنعت نیمه‌هادی، روند کوچک‌سازی افزارهای کنونی مبتنی بر عنصر سیلیکون با چالش جدی و غیر قابل گزین روبرو است. بنابراین تحقیقات گستردگی‌های در زمینه‌ی رفع این چالش‌ها صورت پذیرفته و می‌پذیرد. براساس تحقیقات، افزارهای مبتنی بر عنصر کربن توانسته‌اند جایگاه ویژه‌ای در این عرصه پیدا کنند.

در این مقاله سعی شده است تا با استفاده از یکی از ساختارهای معروفی شده‌ی مبتنی بر کربن، به نام گرافن، طراحی و مشخصه‌یابی یک بلوک منطقی قابل پیکربندی (CLB)، که در تراسه‌های FPGA مورد استفاده است، صورت پذیرد. البته ساختار داخلی این واحد به گونه‌ای است که طراحی افزارهای متداول و پرکاربرد در طراحی دیجیتال، همچون: فلیپ‌فلاب نوع-D (DFF) و لچ (Latch) را نیز در پی دارد. سرعت حدود ۵۸ برابری در مقایسه DFF مبتنی بر سیلیکون و طراحی و استخراج پارامترهای زمانی مدار CLB و المان‌های داخلی تمام گرافنی آن نتایجی است که در این مقاله به آن پرداخته خواهد شد.

کلمات کلیدی: بلوک منطقی قابل پیکربندی، گرافن، فلیپ‌فلاب، پارامترهای زمانی، گیت قابل پیکربندی مجدد.

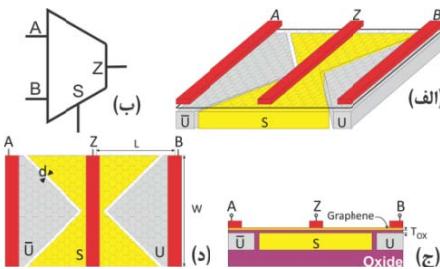
۱- مقدمه

الکترونیک نوری، از جمله مواد و روش‌های معرفی شده برای جایگزین سیلیکون می‌باشند [۶-۲].

مواد مبتنی بر کربن بعد از معرفی ساختار پایدار دو بعدی آن در سال ۲۰۰۴ میلادی، توجه بسیاری را به خود جذب کرد. در این میان ساختارهای نانو-لوله کربنی و گرافن دارای ویژگی‌های الکتریکی بسیار کاربردی برای استفاده در الکترونیک هستند. نانو-لوله کربن به علت مشکلات مرحله ساخت، نگاه‌ها را به سمت استفاده از گرافن تغییر داده است. گرافن به شبکه‌ای شش ضلعی از اتم‌های کربن که در یک صفحه قرار می‌گیرند، گفته می‌شود. افزارهای مبتنی بر این نوع ساختارها از لحاظ عملکردی نسبت به ترانزیستورهای معمول متفاوت عمل خواهند کرد، از این رو نیاز است تا مدارهایی که طراحی آن بر پایه این ساختارها صورت می‌پذیرد، مورد مطالعه قرار گرفته و مشخصه‌یابی و بررسی شوند. شکاف باند الکتریکی در گرافن صفر است، این مورد یک چالش بسیار بزرگ برای استفاده از آن در صنایع الکترونیکی محسوب می‌شود، چرا که رفتار شبه فلزی دارد و حالت

صنعت نیمه‌هادی و الکترونیک در چند دهه گذشته، رشد فزاینده در تراسه تراسه‌ها را مطالبه کرده است، که این عمل نیاز به کوچک‌سازی المان‌های پایه‌ای یعنی همان ترانزیستورها دارد. این کوچک‌سازی طبق آنچه به نام قانون مور شناخته می‌شود، منجر به افزایش تراکم دو برابری ترانزیستورها در تراسه‌ها طی هر دو سال شده است [۱]. طبق پیش‌بینی‌ها انتظار می‌رود این روند حداقل تا چند سال آینده و سرانجام عنصر سیلیکون قابل کوچک‌سازی بیشتر نباشد [۲]. برای غلبه بر این محدودیتها و بر طرف کردن نگرانی‌های صنعت، تحقیقات گستردگی‌های در رابطه با مواد فرا-سیلیکونی برای جایگزین کردن با سیلیکون، انجام شده است. ترانزیستورهای مبتنی بر نانو-لوله کربنی، الکترونیک مولکولی، اسپینtronیک، افزارهای مبتنی بر محاسبات کوانتومی، افزارهای مبتنی بر حافظه مغناطیسی و

(پایانه‌های U، S و Z) جدا می‌شود. پایانه‌های فلزی (پایانه‌های A، B و Z) نیز بر روی صفحه گرافن قرار می‌گیرند. در شکل ۱ پارامترهای w ، d و T_{ox} به ترتیب پارامتر طراحی مربوط به اندازه افزاره، کمینه اندازه (فاصله پایانه‌ها) در افزاره و ضخامت لایه اکسید در افزاره گرافنی هستند. با تنظیم و طراحی پارامتر w می‌توان خازن معادل در افزاره را نیز کنترل و طراحی نمود [۹، ۱۰]. علاوه بر پارامترهای ساختاری، ولتاژ تغذیه نیز از جمله پارامترهای موثر در طراحی به شمار می‌رود.

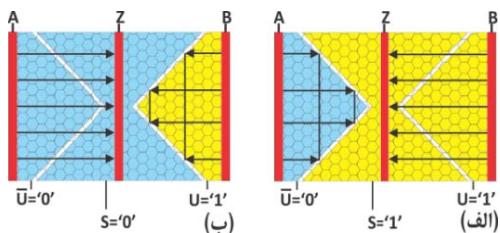


شکل ۱- گیت قابل پیکربندی گرافنی (GRG)، (الف) نمای سه بعدی، (ب) المان معادل GRG، (ج) نمای روبرو، (د) نمای از بالا [۱۱]

با فرض این که ورودی‌های کنترل پیکربندی افزاره U و Ubar، به ترتیب به ولتاژ سطح بالا (یک منطقی) و ولتاژ سطح پایین (صفر منطقی) متصل باشند، توضیح عملکرد افزاره در ادامه آورده شده است. با توجه به شکل ۲ با اعمال مقدار یک منطقی به گیت Z، ناحیه گیت میانی به نوع n تبدیل می‌شود (رنگ زرد)، سپس یک ناحیه p-n در بین پایانه‌های Z و A به وجود می‌آید. مطابق شکل ۲ (الف) باریکه‌های الکترون وقتی به محل پیوند p-n مورب با زاویه ۴۵ درجه، می‌رسند با مقاومت بسیار زیاد R_{p-n} مواجه می‌شوند و اجازه عبور پیدا نمی‌کنند. در سمت دیگر حامل‌های پایانه‌ی B در مسیر حرکت خود به سمت پایانه Z فقط ناحیه n با مقاومت بسیار پایین R_{n-n} را می‌بینند. این حامل‌ها به راحتی از مسیر موجود عبور نموده و به پایانه‌ی Z وارد می‌شوند. در این صورتتابع خروجی Z=B در حالت '1' به دست می‌آید.

با توجه توضیحات ارائه شده، در حالتی که گیت S برابر با '0' منطقی باشد، مطابق شکل ۲ (ب) حامل‌های پایانه‌ی A در مسیر خود به پایانه Z فقط ناحیه p می‌باشد، اینکه می‌تواند مسیر با مقاومت پایین R_{p-p} را می‌بینند، و نهایتاً تابع خروجی Z=A در حالت '0' به دست می‌آید. به صورت خلاصه، اگر از دید مدار منطقی به این افزاره بنگریم، المان معروفی شده عملکردی شبیه به مالتی‌پلکسor با تابع منطقی $Z=((\#S).A)+(S.B)$ را ارائه می‌دهد.

با اعمال مقدار مناسب به ورودی‌های A و B و گیت کنترلی S در افزاره معرفی شده، انواع تابع دو ورودی منطقی قابل پیاده‌سازی می‌باشد. از این موارد می‌توان به معکوس‌کننده و گیت‌های منطقی AND و OR اشاره کرد. حالت‌ها و توابع گوناگون قابل پیاده‌سازی دیگر در جدول ۱ قابل مشاهده است.



شکل ۲- عملکرد GRG، (الف) وقتی $S=1'$ است، (ب) عملکرد وقتی $S=0'$ است [۱۱]

خاموش و روشن یک افزاره با شکاف باند صفر قابل تشخیص نیست [۷، ۸]. برای حل مشکل شکاف باند صفر، راه کارهایی از قبیل نانو-نوار گرافنی و ساختار گیت‌های مجزا ارائه شده است. استفاده از نانو-نوارهای گرافنی به عملکرد افزاره‌های مبتنی بر آن واپسگی شدید به فرآیند ساخت آن دارد [۹]. یک روش دیگر برای استفاده از گرافن، بهره‌گیری از ایجاد ناخالصی توسط اعمال ولتاژ الکترواستاتیک به آن است. در این حالت با اعمال ولتاژ الکترواستاتیک به پایانه‌های قرار گرفته در پشت صفحه گرافن، تراز فرمی آن تغییر می‌کند [۱۰، ۹]. ایجاد ناخالصی الکترواستاتیکی دروازه‌های جدیدی را برای دستیابی به افزاره‌های مبتنی بر گرافن و نانو-الکترونیک و سیستم‌های مبتنی بر گرافن گشوده است. افزاره‌های مبتنی بر این نوع ساختارها از لحاظ عملکردی نسبت به ترانزیستورهای معمول متفاوت عمل خواهند کرد، از این رو نیاز است تا مدارهایی که طراحی آن بر پایه این ساختارها صورت می‌پذیرد، مورد مطالعه قرار گرفته و مشخصه‌یابی و بررسی شوند.

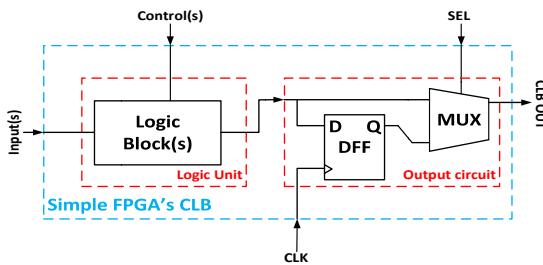
گیت قابل پیکربندی گرافنی (GRG)، یکی از المان‌هایی است که با استفاده از روش اخیر، طراحی شده است. مراجع [۱۲، ۱۱] مدل‌هایی برای بررسی پارامترهای زمانی و توان مصرفی در گیت‌های پایه مبتنی بر این ساختار را ارائه داده‌اند، اما هیچ یک به بررسی، طراحی و ارائه مدلی برای المان‌های ترتیبی، که از موارد ضروری و مورد نیاز در مدارهای دیجیتال هستند، نپرداخته‌اند. با در نظر گرفتن نیاز اساسی سامانه‌های دیجیتال به المان‌های ترتیبی، لازم است تا المان‌های حافظه دار پایه، طراحی و مشخصه‌یابی شوند. لذا در این مقاله طراحی لج و فلیپ‌فلاب و مشخصه‌یابی آن در راستای طراحی و مشخصه‌یابی CLB ارائه خواهد شد.

در ادامه روند ارائه مطالب به این صورت است. در بخش (۲) در رابطه با ساختار مورد استفاده در این مطالعه و نحوه عملکرد آن توضیح داده می‌شود و در بخش (۳) روند طراحی بلوک قابل پیکربندی همراه با طرح مداری مربوط به المان‌های داخلی مورد نیاز آن ارائه می‌شود. همچنین فرآیند و شرایط آزمون‌های انجام شده و نیز مراحل مشخصه‌یابی المان‌های پیشنهادی در بخش (۴) ارائه خواهد شد. در نهایت، در بخش (۵) نتیجه‌گیری و خلاصه‌ای از مباحث مطرح شده، ارائه شده است.

۲- گیت قابل پیکربندی گرافنی

طراحی المان GRG به صورت فیزیکی و شماتیک در شکل ۱ نمایش داده شده است [۱۰، ۹]. این المان دارای شش پایانه کتریکی می‌باشد. با توجه به شکل ۱ پایانه‌های کتریکی GRG عبارتند از: دو پایانه‌ی U و Ubar که وظیفه ایجاد ناخالصی الکترواستاتیکی و کنترل پیکربندی افزاره را دارند. بدین صورت که وقتی ولتاژی مثبت به یکی از این پایانه‌ها اعمال می‌شود قسمت گرافنی روی آن به مانند وقتی عمل می‌کند که ناخالصی نوع n به آن وارد شده باشد. با توجه به اینکه دو پایانه در تمام شرایط منطق متفاوت از هم دارند، در حالت فرض شده، پایانه دیگر دارای ولتاژ منفی خواهد بود و ناحیه گرافنی واقع در روی آن مطابق زمانی رفتار خواهد کرد که ناخالصی نوع p به آن وارد شده باشد. بنابراین با توجه به اینکه این نوع ناخالصی از منشا الکترواستاتیکی به وجود می‌آید نوع آن را ناخالصی الکترواستاتیکی است. همچنین پایانه‌های A، B و S به عنوان ورودی‌های المان و پایانه Z نیز خروجی خواهد بود. ورودی S در این المان نیز زیر صفحه گرافنی قرار دارد که متعاقباً همانند پایانه‌های U و Ubar ناخالصی‌های الکترواستاتیکی متضاد را ایجاد خواهد نمود. در گیت قابل پیکربندی گرافنی صفحه گرافن به وسیله‌ی یک لایه اکسید یا به عبارت دیگر یک لایه عایق از پایانه‌های زیرین خود

استفاده است، مشخص نمود. در شکل ۴ بخش اول با نام Logic Unit و بخش دوم نیز با نام Output Circuit قابل مشاهده است. در ادامه به بررسی روند طراحی این بخش‌ها پرداخته می‌شود.

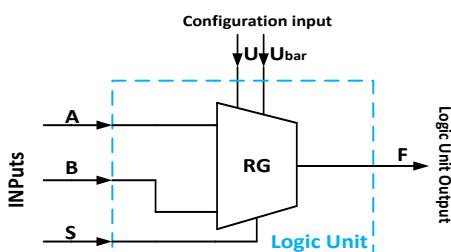


شکل ۴- طرح پایه برای یک CLB

۱-۳ واحد منطقی^۴

وظیفه واحد منطق فراهم آوردن عملکرد توابع منطقی مختلف در کنار ارائه قابلیت پیکربندی است. به عنوان مثال با توجه به شکل ۴ با تعیین سیگنال‌های (Control(s)، می‌توان تابع منطقی مانند: AND، OR و NOT را به ورودی (های) (Control(s)) اعمال نمود و نتیجه را در خروجی واحد منطق دریافت کرد.

یکی از امان‌هایی که می‌تواند عملکرد اشاره شده را تحقق بخشد، یک مالتی‌پلکسor است. همانطور که در بخش قبل اشاره شده، یک گیت قابل پیکربندی گرافنی به طور طبیعی می‌تواند عملکردی شبیه یک مالتی‌پلکسor را در اختیار قرار می‌دهد. در کنار این خصوصیت می‌توان این مورد را نیز اضافه کرد که با اعمال مناسب سیگنال به ورودی‌های کنترل یک GRG می‌توان پیکربندی افزاره را نیز تغییر داد (رجوع شود به جدول ۱). این مورد باعث می‌شود که خصوصیت قابلیت پیکربندی مجدد نیز به افزاره نهایی اضافه شود. بنابراین طرح مداری واحد منطق به شکل ۵ طراحی و پیشنهاد شده است.



شکل ۵- طرح مداری پیشنهادی جهت دستیابی به واحد منطق مبتنی بر GRG

۲-۳ مدار خروجی^۵

مدار خروجی کارکرد و وظیفه‌ای یکسان در طراحی CLB‌ها دارد و از ایده‌ای ثابت پیروی می‌کند. این بخش با توجه به شکل ۶ باید بتواند با اعمال سیگنال متناسب به ورودی انتخاب (SEL) در دو حالت متفاوت قرار گیرد. بدین صورت که در یک حالت به صورت مستقیم ورودی (D) را به خروجی انتقال دهد. در حالت دیگر، این بخش باید قادر باشد که ورودی (D) پس از دریافت پالس ساعت (CLK) به خروجی انتقال دهد. به عبارت دیگر در حالت دوم ورودی پس از زمان مشخصی و معین در خروجی ظاهر می‌شود.

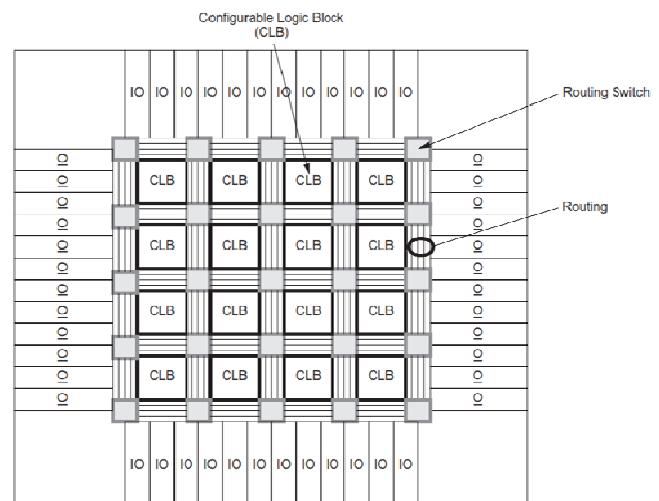
از مزایای این ساختار می‌توان به قابلیت پیکربندی مجدد آن و تغییرتابع منطقی با تغییر ولتاژ گیت‌های U و Ubar اشاره نمود. این خاصیت باعث می‌شود که حتی حین کار بتوان تابع عملکرد این المان را، با تعویض ولتاژ پایانه‌های کنترلی، تغییر داد. همان‌طور که در جدول ۱ مشاهده می‌شود، چهارتابع توسط 'U=1' ایجاد می‌شود؛ در این حین با تغییر به 'U=0' در تابع ایجاد شده S به 'S' تبدیل می‌شود و تابع جدید ایجاد می‌گردد.

جدول ۱- تابع حاصل از ورودی‌های مختلف در GRG [۱۰، ۱۲]

MUX	INV	AND	OR	$f = ?$
$\begin{array}{c} x \\ y \\ \hline A \quad B \quad S \\ q \end{array}$ $f = q' \cdot x + q \cdot y$	$\begin{array}{c} 1 \\ 0 \\ \hline A \quad B \quad S \\ q \end{array}$ $f = q'$	$\begin{array}{c} 0 \\ y \\ \hline A \quad B \quad S \\ q \end{array}$ $f = q \cdot y$	$\begin{array}{c} x \\ 1 \\ \hline A \quad B \quad S \\ q \end{array}$ $f = x + q$	$\begin{array}{c} f \\ \hline \end{array}$
$\begin{array}{c} x \\ y \\ \hline A \quad B \quad S \\ q \end{array}$ $f = q \cdot x + q' \cdot y$	$\begin{array}{c} 1 \\ 0 \\ \hline A \quad B \quad S \\ q \end{array}$ $f = q$	$\begin{array}{c} 0 \\ y \\ \hline A \quad B \quad S \\ q \end{array}$ $f = q' \cdot y \equiv (q + y')'$	$\begin{array}{c} x \\ 1 \\ \hline A \quad B \quad S \\ q \end{array}$ $f = x + q \equiv (q, x)'$	$\begin{array}{c} f \\ \hline \end{array}$
MUX	BUFFER	AND	OR	

۳- طراحی بلوك منطقی قابل پیکربندی

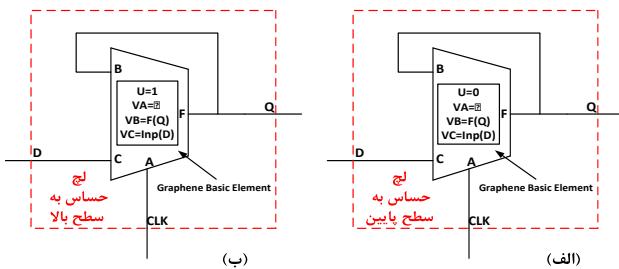
به طور کلی مطابق شکل ۳ اجزاء یک تراشه FPGA به سه قسمت تقسیم می‌شود که عبارتند از: سیم‌کشی^۶، کلیدهای انتخاب سیم‌کشی^۷ و بلوك منطقی قابل پیکربندی (CLB). با توجه به ارائه گزارش‌هایی در مورد سیم‌کشی و کلید در طراحی FPGA مبتنی بر گرافنی [۹]، نیاز به یک CLB تمام گرافنی نیز برای این بخش است. با در نظر گرفتن عدم ارائه گزارشی مبنی بر طراحی CLB براساس المان GRG، در ادامه این تحقیق برای اولین بار به طراحی و مشخصه‌یابی CLB تمام گرافنی مبتنی بر GRG می‌پردازد.



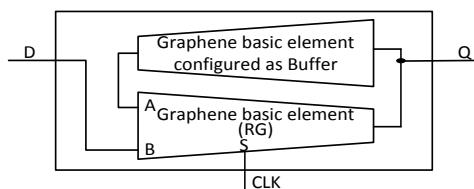
شکل ۳- طرح نمونه از داخل تراشه FPGA

بطور کلی یک CLB از دو بخش تشکیل می‌شود؛ بخش اول، واحد منطقی است که شامل مدار قابل پیکربندی و تولید کننده تابع منطقی است. بخش دوم نیز مدار خروجی است که با کنترل ورودی‌های آن می‌توان نوع خروجی را به صورت خروجی مستقیم (ترکیبی)، که در طراحی مدارهای ترکیبی مورد استفاده است، یا خروجی زمانبندی شده (ترتیبی)، که در طراحی مدارهای ترتیبی مورد

دست‌یابی به یک لج مقاوم در برابر اثر بارگذاری طبقات بعد، طرح مفهومی نهایی با خروجی بهینه شده برای تغذیه چهار معکوس کننده به عنوان ورودی طبقه بعد $FO4^7$ ، توسط یک بافر مطابق شکل ۹ ارائه شد.



شکل ۸- طرح پیشنهادی برای لج، (الف) لج حساس به سطح پایین، (ب) لج حساس به سطح بالا



شکل ۹- طرح نهایی لج با خروجی مقاوم شده در برابر اثر بارگذاری طبقه بعد

۴- مشخصه‌یابی بلوک منطقی قابل پیکربندی

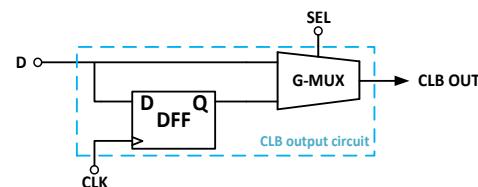
گیت پایه گرافنی یا به عبارت دیگر GRG (Verilog-A) توسط مراجع [۱۱، ۹] ارائه شده است. از ویژگی این مدل می‌توان به قابلیت شبیه‌سازی آن در محیط‌های شبیه‌سازی مداری مانند HSPICE اشاره کرد. در این تحقیق نیز با استفاده از نرم‌افزار HSPICE و مدل Verilog-A شبیه‌سازی‌ها انجام شده است. لازم است اشاره شود که در تمام شبیه‌سازی‌ها از دمای ۲۵ درجه سانتیگراد استفاده شده است.

۱-۴- روش آزمون و بررسی افزارهای پیشنهادی

برای بدست آوردن مشخصه‌های زمانی مدار مورد تست (C.U.T.)^۸ طراحی شده و بررسی عملکرد افزارهای پیشنهادی از ساختار مداری مطابق شکل ۱۰ استفاده شده است. در این مدار هر یک از ورودی‌ها از دو گیت معکوس کننده عبور کرده و به افزاره اعمال شده است. این عمل باعث می‌شود تا رفتار ورودی در شبیه‌سازی‌ها از حالت ایده‌آل غیرواقعی به یک حالت واقعی و طبیعی برسد و نتایج نیز به آنچه در واقعیت رخ خواهد داد نزدیکتر شود [۱۳]. این مورد را می‌توان از رفتار سیگنال ورودی بعد از عبور از دو معکوس کننده در شکل ۱۱ مشاهده کرده، مطابق این شکل لبه‌های تیز سیگنال ورودی که ناشی از رفتار ایده‌آل منابع تعریف شده در نرم‌افزار شبیه‌ساز است به لبه‌های نرم و مشابه آنچه که در واقعیت مشاهده می‌شود تبدیل شده است.

به علاوه این ساختار، در خروجی مدار مورد آزمایش نیز بار $FO4$ اعمال می‌نماید تا عملکرد افزارهی مورد آزمایش را در شرایط مداری با بارگذاری طبقات بعدی مورد بررسی قرار دهد. با اعمال بار $FO4$ که معادل ورودی چهار گیت معکوس کننده است می‌توان از عملکرد صحیح افزاره در اکثر شرایط بارگذاری مداری، اطمینان حاصل نمود. مطابق آنچه اشاره شد که در این مطالعه،

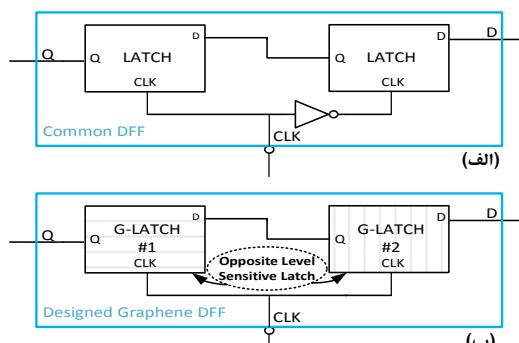
همانطور که در شکل ۶ مشاهده می‌شود، در مدار خروجی از DFF به عنوان المان زمانبندی استفاده شده است که در ادامه روند طراحی آن به عنوان مدار داخلی بخش مدار خروجی، ارائه می‌شود. این المان وظیفه نگهداری مقدار قبلی را تا قبل از اعمال پالس ساعت دارد و پس از رخ داد پالس ساعت مقدار جدید را به ورودی مالتی‌پلکسر خروجی جهت ارسال به خروجی CLB ارائه می‌کند. با این عمل خروجی نهایی یک خروجی زمان بندی شده خواهد بود.



شکل ۶- طرح مداری پیشنهادی جهت دست‌یابی به مدار خروجی CLB

۳-۳- طراحی DFF گرافنی

برای دست‌یابی به یک فلیپ‌فلاپ نوع-D، در میان طرح‌های مداری موجود در طراحی ASIC^۹، استفاده از لجهای پایه و پیرو طرحی پذیرفته شده است [۱]. طرح پایه DFF در شکل ۷ (الف) قابل مشاهده است. جهت دست‌یابی به DFF نیاز است که دو لج با حساسیت به سطح متفاوت به صورت سری قرار گیرند. در حالت معمول نیاز است که یا دو نوع لج با حساسیت به سطح متفاوت طراحی شوند و یا اینکه از بین دو لج یکسان، در ورودی یک لج، یک گیت معکوس کننده قرار گیرد تا حساسیت به سطح تغییر کند. اما یکی از مزایای لج گرافنی که در ادامه معرفی می‌شود، این است که تنها با اعمال ولتاژ متفاوت به پایانه پیکربندی آن، حساسیت به سطح تغییر خواهد کرد. بنابراین طرح نهایی برای فلیپ‌فلاپ پیشنهادی به شکل ۷ (ب) خواهد بود. از مزایای این طرح کاهش مصرف توان و سطح مصرفی تراشه و افزایش سرعت افزاره است. به علاوه، اثر بارگذاری بر روی شبکه پالس ساعت نیز کاهش می‌یابد که امری بسیار مهم در طراحی تراشه‌های با پیچیدگی بالا همانند FPGA است.

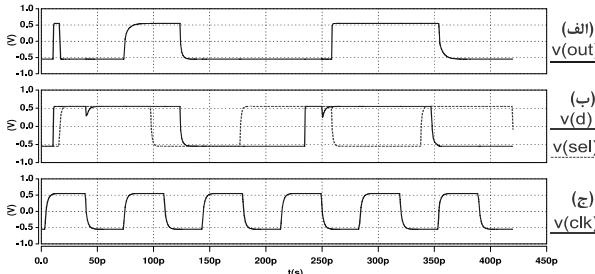


شکل ۷- طرح پایه و پیرو جهت دست‌یابی به DFF، (الف) طرح پایه پیرو متداول، (ب) طرح پایه و پیرو پیشنهادی برای DFF گرافنی

با بهره‌گیری از خاصیت مالتی‌پلکسری گیت پایه گرافنی (GRG)، طرح لج مطابق شکل ۸ پیشنهاد می‌شود. همان‌گونه که در این شکل مشاهده می‌شود با تغییر سیگنال U (و متعاقباً Ubar) در المان گرافنی پایه، حساسیت به سطح که از ویژگی‌های یک لج می‌باشد، تغییر می‌نماید. با استفاده از طرح ارائه شده، جهت

۴-۳- شبیه‌سازی مدار خروجی

قسمت مدار خروجی در طراحی یک CLB در شکل ۶ نشان داده شد. این بخش با در نظر گرفتن شرایط اعمالی به مدار تست که در روش آزمون اشاره شد، مورد آزمون و شبیه‌سازی قرار گرفت. نتایج شبیه‌سازی بخش مدار خروجی در شکل ۱۳ نمایش داده شده است.



شکل ۱۳- نتایج شبیه‌سازی مدار خروجی، (الف) خروجی Q، (ب) ورودی D (خط پیوسته)، ورودی انتخاب SEL (نقطه چین)، (ج) پالس ساعت CLK

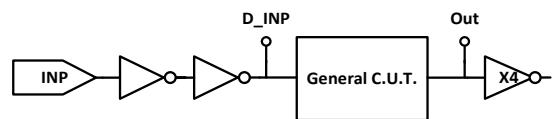
مطابق شکل ۱۳ عملکرد مطلوب در مدار خروجی مشاهده می‌شود. توضیح شکل ۱۳ بدین شرح است که زمانی که خط انتخاب S در وضعیت غیرفعال قرار دارد خروجی وضعیت ورودی را دنبال می‌کند. در حالت دیگر نیز وقتی خط انتخاب در حالت فعال قرار می‌گیرد، خروجی وضعیت ورودی را پس از اعمال پالس ساعت دنبال خواهد کرد. به عنوان یک مثال خوب می‌توان رفتار این مدار را در حوالی زمان صفر در نظر گرفت. در این لحظه ابتدا پالس ساعت اعمال می‌شود بنابراین مقدار فعلی ورودی خود که '۰' هست را ذخیره می‌کند ولی جون مدار در حال عبور مستقیم سیگنال هست تا زمانی که خط sel از حالت '۰' به '۱' منطقی تغییر نکرده است خروجی همان مقدار ورودی است. بعد از تغییر سطح خط sel مقدار ذخیره شده DFF را در خروجی می‌بینیم. این مقدار تا اعمال پالس ساعت بعدی و تغییر خروجی DFF حفظ شده است. توضیحات بیشتر در مورد طراحی DFF در ادامه ذکر می‌شود.

۴-۴- مشخصه‌یابی DFF

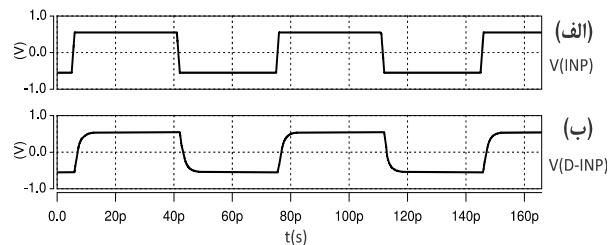
این تحقیق، جهت مشخصه‌یابی و تعیین پارامترهای زمانی فلیپ‌فلاب طراحی شده از روش یافتن ماکریم تاخیر در خروجی نسبت به تاخیر ورودی استفاده نموده است [۱۴].

با استفاده از روش اشاره شده، DFF پیشنهادی مورد بررسی قرار داده شد و با توجه به حالت اولیه خروجی (سطح بالا یا سطح پایین) و نوع لبه ورودی (بالا رونده یا پایین رونده) چهار حالت مختلف جهت استخراج بدترین شرایط ممکن در پارامترهای زمانی مورد آزمون قرار گرفت و نتایج شبیه‌سازی در شکل ۱۴ به نمایش گذاشته شده است. لازم به ذکر است که تنظیم پارامترهای افزاره گرافی منطقی با تکنولوژی ۴۵- نانومتر که توسط ITRS اعلام شده است، صورت پذیرفته است [۲]. با توجه به خلاصه نتایج مشخصه‌یابی زمانی موجود در جدول ۲، فلیپ‌فلاب گرافی (GDFF) سرعتی ۵۸ برابری نسبت به مشابه سیلیکونی موجود خود (CDFF) [۱۵] در تکنولوژی ۴۵- نانومتر نشان می‌دهد. جهت مقایسه افزارهای از لحاظ توان متوسط مصرفی نیز از بار FO4 صرف‌نظر و بار فیزیکی یکسان مقاومتی- خازنی اضافه شد. این تغییر باعث می‌شود که معیاری یکسان برای مقایسه توان مصرفی افزارهای در حضور بار بدست آید تا مقایسه انجام شده از

شبیه‌سازی‌ها با کمک نرم افزار HSPICE انجام شده است. مدل رفتاری گیت پایه گرافی که با استفاده از کدهای Verilog-A توصیف شده است، نیز در حال حاضر در دسترس است و با اضافه شدن این کدها به مفسر داخلی HSPICE، امکان به کارگیری این المان در طراحی‌های مداری و تحلیل آن توسط نرم‌افزار HSPICE ممکن می‌شود.



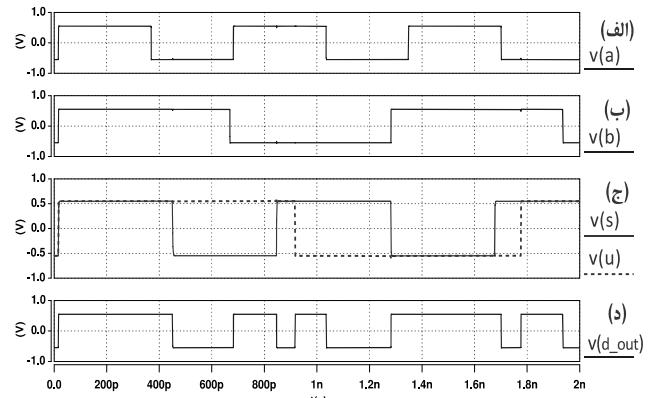
شکل ۱۰- ساختار مداری جهت تست و شبیه‌سازی افزارهای طراحی شده (C.U.T.) مدار تحت آزمایش



شکل ۱۱- رفتار مشابه حالت طبیعی سیگنال ورودی پس از عبور از دو معکوس کننده، (الف) سیگنال ورودی ایده‌آل، (ب) سیگنال واقعی شده ورودی

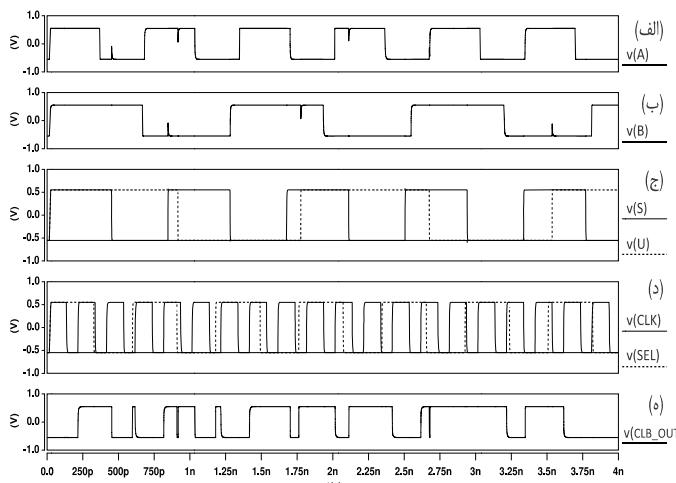
۴-۵- شبیه‌سازی واحد منطق

به علت نوع اتصال این واحد به طبقه بعد مطابق شکل ۴ در این مرحله از اتصال بار FO4 به خروجی واحد منطق برای شبیه‌سازی‌ها صرف‌نظر شده است و تنها ورودی‌ها جهت دست‌یابی به رفتار طبیعی از دو معکوس کننده عبور نموده‌اند. نتایج شبیه‌سازی عملکردی این واحد در شکل ۱۲ ارائه شده است، که مطابق این شکل، رفتار مطلوب از واحد منطق مشاهده می‌شود. به عنوان مثال در زمان ۱ نانوثانیه مقادیر منطقی موجود روی پایانه‌ها بدین شرح است: پایانه‌های a و s مقدار منطقی '۱' و پایانه‌های b و u مقدار منطقی '۰'. بنابراین مطابق عملکرد توضیح داده شده برای این المان در بخش‌های پیشین، مقدار خروجی باید برابر '۱' منطقی باشد که صحت عملکرد این المان در چنین شرایطی با توجه به نتایج شبیه‌سازی در شکل ۱۲ قابل استنباط است.

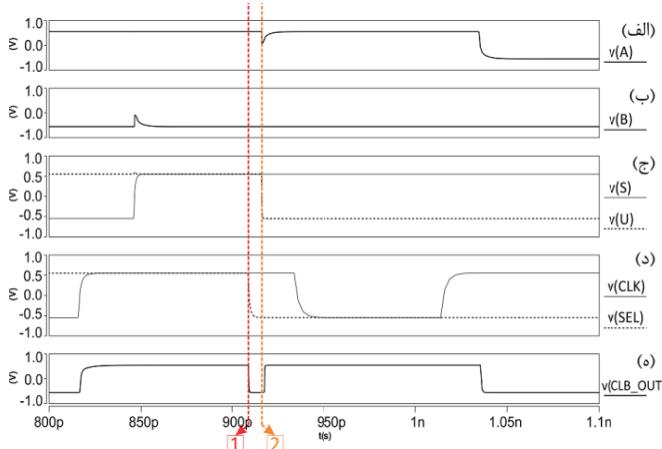


شکل ۱۲- نتایج شبیه‌سازی واحد منطق، (الف) ورودی A، (ب) ورودی B، (ج) ورودی انتخاب S (خط پیوسته)، ورودی U (نقطه چین)، خروجی واحد منطق

مریبوط به CLB با ثابت نگه داشتن مدار خروجی در حالت انتقال مستقیم، مورد بررسی قرار گرفت و نتایج استخراج شد و در جدول ۳ ارائه شده است. به منظور بررسی دقیق‌تر شکل ۱۶ به عنوان نمونه با تمرکز بر روی قسمت مناسبی از آن در شکل ۱۷ به بررسی عملکرد مدار به شرح زیر پرداخته شده است. مطابق شکل ۱۷ در خط زمانی شماره "۱" (خط قرمز رنگ) به لبه پایین رونده خط SEL اشاره شده است در حالیکه خطوط "S" و "U" در سطح یک منطقی قرار دارند. بنابراین با تغییر مقدار SEL به صفر منطقی، خروجی که در حالت عبور مستقیم است باید برابر "B" شود؛ که این حالت به درستی دنبال شده است. در حالت بعدی خط زمانی شماره "۲" (خط نارنجی) تغییر حالت خط "U" را با حفظ حالت قبلی مدار را نشان می‌دهد. به دلیل این تغییر، واحد منطق مقدار "A" را باید به خروجی انتقال دهد. صحت این عملکرد نیز در خروجی به وضوح قابل مشاهده است بعلاوه اینکه تاثیر بارگذاری میرا شونده روی ورودی "A" نیز مشخص است.

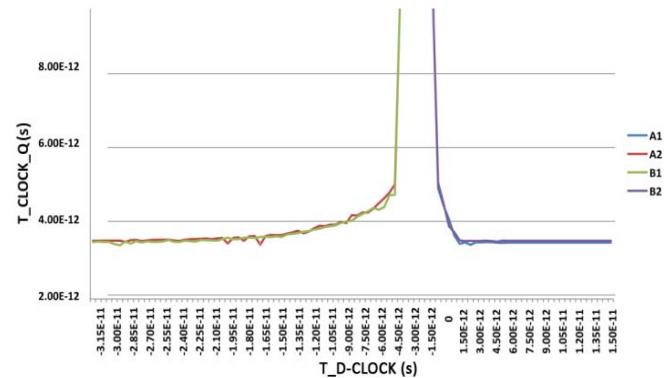


شکل ۱۶- نتایج شبیه‌سازی CLB پیشنهادی، (الف) پیشنهادی، (ب) ورودی A، (ب) ورودی B، (ج) ورودی انتخاب واحد منطق S (خط پیوسته)، ورودی پیکربندی U (نقطه چین)، (د) پالس ساعت CLK (خط پیوسته)، انتخاب خروجی SEL (نقطه چین)، (ه) خروجی نهایی CLB-OUT



شکل ۱۷- بزرگنمایی قسمتی از شکل ۱۶ جهت بررسی دقیق‌تر. نتایج شبیه‌سازی عملکرد CLB پیشنهادی، (الف) ورودی A، (ب) ورودی B، (ج) ورودی انتخاب واحد منطق S (خط پیوسته)، ورودی پیکربندی U (نقطه چین)، (د) پالس ساعت CLK (خط پیوسته)، انتخاب خروجی SEL (نقطه چین)، (ه) خروجی نهایی CLB-OUT

درجه اعتبار و دقت بالاتری برخوردار باشد. در اینجا بار فیزیکی به شرح مقاومت ۱۰ اهمی سری با خروجی و متصل به خازن زمین شده با ظرفیت ۰/۶ فتو فاراد در نظر گرفته شده است. در ادامه برای مشخص شدن شایستگی عملکردی DFF های ارائه شده از پارامتر استاندارد حاصل ضرب تاخیر در توان PDP^۰ استفاده شد. این ضریب شایستگی هر مقدار کوچکتر باشد افزاره سریعتر و کم مصرف تر خواهد بود. با توجه به اطلاعات موجود در جدول ۲ به وضوح دیده می‌شود که GDFF از ضریب شایستگی حدود ۳۱ برابر بهتر از CDFF برخوردار است.



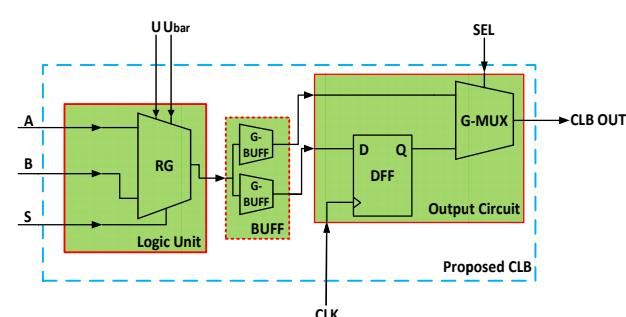
شکل ۱۴- نتایج شبیه‌سازی DFF تحت روش آزمون ماکزیمم تاخیر جهت دستیابی به پارامترهای زمانی آن

جدول ۲- مقایسه پارامترهای زمانی CDFF و GDFF

Power (uW)	Timings (ps)				PDP (W.s)
	Setup time	Hold time	Cont. ^{۱۱} delay	Prop. ^{۱۰} delay	
Graphene	6.88	7.5	-1.5	3.34	33.14
Silicon	3.71	115	-70	251	1042.51
Ratio (Si/G)	0.54	15.33	46.67	75.15	58.3
					31.46

۴- شبیه‌سازی عملکرد و مشخصه‌بایی CLB

پس از آماده سازی دو بخش مورد نیاز CLB با اتصال آنها از طریق دو بافر مطابق شکل ۱۵ به بررسی عملکرد آن توسط شبیه‌سازی حالات ممکن پرداخته شد. شرایط آزمایش شبیه‌سازی در این مورد نیز مطابق روش آزمون اشاره شده تنظیم و طراحی شد و نتایج این آزمون در شکل ۱۶ ارائه شده است. مطابق شکل ۱۶ تمامی حالات ممکن در عملکرد CLB پیشنهادی بررسی شده است و صحبت عملکرد آن مورد اثبات قرار گرفته است.



شکل ۱۵- ساختار CLB پیشنهادی همراه با نام‌گذاری پایانه‌های آن

شبیه‌سازی‌ها برای یافتن مشخصات زمانی CLB پیشنهادی بدین شرح ادامه یافت، که با توجه به در دست داشتن مقادیر تاخیر در DFF، مشخصات زمانی

- [2] ITRS. (2014). *International Technology Roadmap for Semiconductor*. Available: <http://www.itrs2.net/>
- [3] S. Srinivasan, A. Sarkar, B. Behin-Aein, and S. Datta, "All-spin logic device with inbuilt nonreciprocity," *IEEE Trans. Magnetics*, vol. 47, pp. 4026-4032, 2011.
- [4] ITRS, "Emerging research devices," 2005.
- [5] F. Schwierz, "Graphene transistors," *Nature Nanotechnology*, vol. 5, pp. 487-496, 2010.
- [6] A. K. Geim, and K. S. Novoselov, "The rise of graphene," *Nature Materials*, vol. 6, pp. 183-191, 2007.
- [7] Z. Chen, Y.-M. Lin, M. J. Rooks, and P. Avouris, "Graphene nano-ribbon electronics," *Physica E: Low-Dimensional Systems and Nanostructures*, vol. 40 ,pp. 228-232, 2007.
- [8] X. Wang, Y. Ouyang, X. Li, H. Wang, and et. al., "Room-temperature all-semiconducting sub-10-nm graphene nanoribbon field-effect transistors," *Physical Review Letters*, vol. 100, pp. 206803, 2008.
- [9] S. Tanachutiwat, "Graphene-based post-CMOS architecture," State University Of New York At Albany, 2012.
- [10] S. Tanachutiwat, J. Ung Lee, W. Wang, and C. Y. Sung, "Reconfigurable multi-function logic based on graphene pn junctions," *Design Automation Conference (DAC), Anaheim, CA*, pp. 883-888, 2010.
- [11] S. Miryala, M. Montazeri, A. Calimera, E. Macii, and et. al., "A verilog-a model for reconfigurable logic gates based on graphene pn-junctions," in *Proc. of the Conference on Design, Automation and Test in Europe*, pp. 877-880, 2013.
- [12] S. Miryala, A. Calimera, E. Macii, M. Poncino, and et. al., "Investigating the behavior of physical defects in pn-junction based reconfigurable graphene devices," in *14th Latin American Test Workshop (LATW)*, pp. 1-6, 2013.
- [13] V. G. Oklobdzija, V. M. Stojanovic, D. M. Markovic, and N. M. Nedovic, *Digital system clocking: high-performance and low-power aspects*, John Wiley & Sons, 2005.
- [14] N. Nedovic, W. W. Walker, and V. G. Oklobdzija, "A test circuit for measurement of clocked storage element characteristics," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1294-1304, 2004.
- [15] PTM. Predictive Technology Model [Online]. Available: <http://ptm.asu.edu>.

با توجه به مقدار گزارش شده در جدول ۳، مشاهده می‌شود که CLB پیشنهادی در حالت انتقال مستقیم خروجی واحد منطق، در بدترین شرایط ممکن، تاخیری در حدود ۱/۵۴ پیکو-ثانیه از خود نشان می‌دهد. در صورتیکه حالت انتقال زمان بندی شده مد نظر گیرد با تاخیر انتشار DFF (موجود در جدول ۲) را نیز به مقدار اشاره شده اضافه کرد.

جدول ۳- مشخصات زمانی CLB پیشنهادی در حالت انتقال مستقیم خروجی واحد منطق به خروجی CLB

	A	B	S	U	Logic Unit output		CLB_OUT (Direct Mode)	
					min	Max	min	Max
Delay (s)								
A	---	0	0	1	4.58E-13	9.20E-13	6.02E-13	8.02E-13
			1		8.95E-13	1.26E-12	5.00E-13	1.08E-12
		0	1	0	1.22E-12	1.25E-12	1.11E-12	1.14E-12
		1			1.16E-12	6.12E-13	9.54E-13	9.99E-13
B	---	0	0	0	9.68E-13	1.20E-12	6.42E-13	1.01E-12
		1			1.11E-12	1.29E-12	1.05E-12	1.27E-12
		0	1	1	1.11E-12	1.29E-12	1.05E-12	1.27E-12
		1			1.11E-12	1.28E-12	1.04E-12	1.27E-12
S	---	0	1	0	1.53E-12	1.57E-12	1.47E-12	1.51E-12
		1			1.46E-12	1.60E-12	1.46E-12	1.50E-12
		0	0	1	1.55E-12	1.58E-12	1.52E-12	1.54E-12
		1			1.36E-12	1.44E-12	1.23E-12	1.34E-12
U	---	0	1	0	1.42E-12	1.43E-12	1.36E-12	1.40E-12
		1	0		1.52E-12	1.54E-12	1.46E-12	1.51E-12
		0	1	---	1.29E-12	1.41E-12	1.28E-12	1.28E-12
		1	0		1.42E-12	1.43E-12	1.37E-12	1.37E-12

۵- نتیجه‌گیری

در این مقاله با تمرکز بر طراحی بلوک منطقی قابل پیکربندی، طراحی آن در سطح مداری ارائه شد و مورد بررسی قرار گرفت. این تحقیق قابلیت به کارگیری گیت قابل پیکربندی گرافنی را در طراحی‌های مداری نشان داد، به علاوه اینکه از پایانه‌ی پیکربندی این گیت به عنوان یک ورودی استفاده شد تا آزادی عمل بیشتر و بهتری برای دستیابی به یک CLB حاصل آید.

در ادامه، طراحی مدارهای داخلی CLB از جمله Latch و DFF در این مطالعه ارائه شد و بهینه‌سازی و مشخصه‌یابی لازم انجام پذیرفت تا علاوه بر استفاده این المان‌ها در مدار داخلی CLB، بتوان از این المان‌ها به صورت جداگانه نیز استفاده کرد.

بلوک منطقی قابل پیکربندی ارائه شده در این مطالعه، سرعتی به مراتب بالاتر نسبت به مشابه سیلیکونی خود نشان داد، چرا که گره حیاتی آن که انتظار بیشترین تاخیر در آن می‌رود، یعنی DFF سرعتی حدود ۵۸ برابر بیشتر از مشابه خود در تکنولوژی سیلیکونی نشان می‌دهد.

سپاسگزاری

مولفین بر خود لازم می‌دانند که از همکاری علمی جناب آقای دکتر سندیپ میریالا (Sandeep Miryala) از موسسه ملی فیزیک زیر اتمی (Nikhef) هلند، تشکر و قدردانی به عمل آورند.

مراجع

- [1] N. Weste and D. Harris, *CMOS Vlsi Design*, 4th ed. Boston, MA: Pearson Addison Wesley, 2011.

سیدعلی سیف کاشانی دانشآموخته در مقاطع



کارشناسی و کارشناسی ارشد از دانشگاه کاشان در رشته مهندسی برق با گرایش الکترونیک است. علاقه‌وی در زمینه‌های تحقیقاتی الکترونیک دیجیتال، ساخت و طراحی مدار و بوردهای الکترونیکی و برنامه‌سازی مدارهای کنترلی است. کسب رتبه برتر در مقاطع کارشناسی و کارشناسی ارشد به همراه اتخاذ عنوانی برتر مسابقات ملی جز افتخارات علمی و پژوهشی وی می‌باشد. وی از سال ۱۳۹۲ به تحقیقات در حوزه مواد و ساختارهای نوین فراسیلیکنی و گرافن پرداخته است و نتایج این تحقیقات در قالب مقالات در کنفرانس‌ها و نشریات داخلی و خارجی در دسترس قرار گرفته است.

آدرس پست الکترونیکی ایشان عبارت است از:

s.a.seif.k@grad.kashanu.ac.ir

حسین کریمیان علی‌داش مدرک کارشناسی ارشد و



دکتری خود را در رشته مهندسی برق، گرایش الکترونیک به ترتیب در سالهای ۱۳۸۰ و ۱۳۸۹ از دانشگاه صنعتی اصفهان اخذ کرده و از سال ۱۳۸۹ و پس از طی یک دوره پسادکتری عضو هیئت علمی دانشگاه کاشان است. حوزه تخصصی فعالیت‌های پژوهشی ایشان طراحی سیستم‌ها و مدارهای دیجیتال کم‌صرف، قابلیت اطمینان، ساختارها و مواد نوظهور، مدارهای پردازش تصویر و همچنین منابع تغذیه و کاربردهای صنعتی است. در این زمینه‌ها، تاکنون مقالات متعددی در مجلات و کنفرانس‌های معتبر داخلی و خارجی از ایشان به چاپ رسیده است.

آدرس پست الکترونیکی ایشان عبارت است از:

hkarimian@kashanu.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۵/۰۴/۰۲

تاریخ اصلاح: ۱۳۹۵/۰۵/۱۴

تاریخ قبول شدن: ۱۳۹۵/۰۶/۱۳

نویسنده مرتبط: دکتر حسین کریمیان علی‌داش، دانشکده برق و کامپیوتر،
دانشگاه کاشان، کاشان، ایران.

¹Graphene Re-Configurable Gate

²Routing

³Switches

⁴Logic Unit

⁵Output Circuit

⁶Application-Specific Integrated Circuit

⁷Fan-Out of Four

⁸Circuit Under Test

⁹Power Delay Product

¹⁰Propagation

¹¹Contamination



نسخه نهائی مقالات ارسالی برای چاپ در نشریه "علوم رایانش و فناوری اطلاعات" باید بر طبق اصول مطرح شده در این راهنمای تهیه شده باشد. رعایت این اصول در نسخه اولیه نیز قویاً توصیه می‌شود. مقالات به زبان فارسی ارسال گردد.

۱- ساختار مقاله

- عنوان: کوتاه و معرف محتوای مقاله باشد و از ۱۵ کلمه تجاوز نکند.
- نام نویسنده‌گان و مؤسسه محل اشتغال آنان: از ذکر عناوین خودداری شود.
- چکیده فارسی: حاوی تعریف مسأله، روش حل، و نتایج مهم باشد و از ۱۵۰ کلمه تجاوز نکند.
- واژه‌های کلیدی: حداقل ۱۰ کلمه
- بدنه اصلی مقاله: بدنه اصلی با "مقدمه" شروع و با "نتیجه‌گیری" خاتمه می‌یابد. بخش‌ها و زیربخش‌های بدنه اصلی باید شماره‌گذاری شوند.
شماره "مقدمه" یک خواهد بود.
- تشرک و قدردانی (در صورت نیاز).
- مراجع: مراجع به ترتیبی که در متن به آنها رجوع می‌شود آورده شوند. نام مؤلفان مراجع در صورت لزوم در متن بصورت فارسی آورده شود. رجوع به مراجع با ذکر شماره آنها در داخل کروشه ([]) انجام پذیرد.
- پیوست‌ها (در صورت نیاز)
- واژه‌نامه (در صورت نیاز)
- برای مقالات فارسی، عنوان مقاله، نام نویسنده‌گان، مؤسسه محل اشتغال، چکیده، و کلمات کلیدی به زبان انگلیسی نیز در صفحه‌ای جداگانه داده شود.
- بیوگرافی کامل نویسنده‌گان به زبان فارسی به همراه عکس

۲- معادله‌ها، شکل‌ها، جدول‌ها، و عکس‌ها

- معادله‌ها باید با فاصله کافی از بالا و پائین تایپ و به صورت متوالی شماره‌گذاری شوند. شماره معادله در پرانتز در انتهای سمت راست سطر حاوی معادله قرار داده شود. معادلات دستنویس به هیچ شکل قابل قبول نیستند.
- شکل‌ها و جدول‌ها باید دارای شماره و عنوان باشند. در شکل‌ها شماره و عنوان در زیر شکل و در جدول‌ها در بالای شکل قرار می‌گیرد. اعداد و متون روی شکل‌ها و جدول‌ها باید دارای اندازه مناسب و کاملاً خوانا باشند.
- اعداد و کلمات روی شکل‌ها و جدول‌ها در مقالات فارسی به زبان فارسی باشند.
- عکس‌ها سیاه و سفید، برقی، و با کیفیت عالی باشند.

۳- نحوه نگارش مراجع

در لیست مراجع انواع مختلف مرجع‌ها به شکل زیر نوشته شوند:

- [۱] ب. مقدم، ا. تقیوی، و ن. طاهری، آشنایی با شبکه‌های کامپیوتری، چاپ دوم، انتشارات نصر، تهران، ۱۳۷۵.
- [۲] ی. براون، مقدمه‌ای بر شبکه‌های عصبی، ترجمه م. ع. آرام، انتشارات فجر، مشهد، ۱۳۷۰.
- [۳] راهنمای کاربران حسابرس، شرکت پردازش رایانه‌ای ایران، تهران، ۱۳۶۵.

[۴] ج. عارف، استنتاج فازی بوسیله شبکه‌های عصبی، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شهر، ۱۳۷۴.

[۵] ج. حسینی، و ح. ربانی، "تشخیص چهره انسان در تصویر،" نشریه امیرکبیر، سال هشتم، شماره ۴۲، ص ۱۲۵-۱۴۷، ۱۳۷۷.

[۶] ج. حسینی، و ح. ربانی، "تشخیص چهره انسان در تصویر،" در مجموعه مقالات هفتمین کنفرانس سالانه انجمن کامپیوتر ایران، ص ۲۲۴-۲۳۲، ۱۳۸۰.

[۷] M. A. Ahmadi, and M. H. Rahimi, *Fuzzy Set Theory*, New Jersey: Prentice-Hall, 1995.

[۸] M. A. Ahmadi, M. H. Rahimi, and A. Fatemi, "Evidence-Based Recognition of 3D Objects," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 12, no. 10, pp. 811-835, 1994.

[۹] A. Taheri, "On-Line Fingerprint Verification," *Proc. IEEE Intl Conf. Pattern Recognition*, pp. 752-758, 1992.

[۱۰] M. A. Ahmadi, *On-line Fingerprint Verification*, Ph. D. Dissertation, MIT, Cambridge, MA, 1982.

[۱۱] A. J. Washington, "The Fingerprint of MalcomX," <http://www.dermatoglyphics.com>, June 2003.

[۱۲] International Biometrics Group, <http://www.biometricgroup.com>, May 2003.

۴- واژه‌نامه

هر واژه خارجی در واژه‌نامه در انتهای مقاله با شماره‌ای مشخص شود و شماره در معادل فارسی آن واژه در متن، بصورت بالاترین آورده شود.

۵- آماده‌سازی مقاله

- مقاله را با نرم‌افزار Word تایپ نمائید.
- متن چکیده به صورت تک ستونی با طول سطر ۱۸ سانتیمتر و متن مقاله به صورت دو ستون ۸۷ میلیمتر و فاصله دو ستون ۶ میلیمتر تایپ شود. حاشیه‌ها از بالا و پائین برابر ۲۰ میلیمتر و از طرفین برابر ۱۵ میلیمتر اختیار شود.
- فاصله عنوان مقاله در صفحه اول از بالای صفحه برابر ۸۵ میلیمتر باشد و عنوان وسط چین شود.
- کلیه عنوان‌ین بصورت پررنگ با قلم "B Nazanin" تایپ شوند، اندازه قلم عنوان مقاله ۱۸، عنوان سطح اول ۱۴، و عنوان سطح دوم و سوم ۱۲ انتخاب شوند.
- متن چکیده‌ها با قلم "B Nazanin" اندازه ۹، متن مقاله با قلم "Times New Roman" اندازه ۱۰، و کلمات و متن انگلیسی با قلم Roman اندازه ۹ تایپ شوند.
- تمام متن بصورت تک فاصله تایپ شود. اسامی نویسنده‌گان از عنوان مقاله و اسامی نویسنده‌گان از عنوان محل اشتغال نویسنده‌گان دو خط فاصله داشته باشد. بالای هر عنوان یک سطر فاصله قرار داده شود.
- سعی شود تعداد صفحات مقاله از ۳۰ صفحه بیشتر نباشد.

۶- نحوه ارسال مقاله

- ارسال مقاله فقط از طریق ایمیل مجله (csitjour@gmail.com) انجام شود.
- مقاله ارسالی برای نشریه علوم رایانش و فناوری اطلاعات نباید در جای دیگری به چاپ رسیده باشد و یا در زمان بررسی توسط نشریه برای چاپ به نشریه دیگری ارسال گردد.
- پس از قبول مقاله، نسخه نهائی تصحیح شده مقاله باید در قالب‌های Word و PDF به نشریه ارسال گردد.
- در نسخه نهائی باید بیوگرافی کلیه نویسنده‌گان (به زبان فارسی) و عکس آنها در انتهای مقاله قرار داده شود، همچنین عنوان مقاله، نام نویسنده‌گان، مؤسسه محل اشتغال، چکیده، کلمات کلیدی به زبان انگلیسی در فایل جداگانه ارسال شود.

Design and Characterization of All Graphene Configurable Logic Block for FPGA Applications

Sayed Ali Seif Kashani

Hossein Karimiyan Alidash

Department of Electrical and Computer, Kashan University, Kashan, Iran

ABSTRACT

Scaling and making faster devices are the crucial need in developing technology. The International Technology Roadmap for Semiconductor (ITRS) on the other hand predicts some serious and severe challenges in scaling process of Silicon based devices. Hence, to overcome these challenges, the researchers are actively searching for the new methods and materials. Based on the researches, Carbon based devices are dominant candidate for Silicon based ones.

In this paper Graphene, the Carbon based structure, is chosen to design and characterize the Configurable Logic Block (CLB) as the part of a Field Programmable Gate Arrays (FPGAs). Additionally, CLB's circuit design needs some useful logic devices such as: D-Flip Flop (DFF) and Latch. Following paper presents design and timing characterization of CLB and all of its internal devices and then shows that Graphene based DFF presents about 58 times faster response than its Silicon based counterpart.

Keywords: Configurable Logic Block (CLB), Graphene, Flip Flop, Timing Parameter, Reconfigurable Gate.

A Low-Power Redundant Decimal Adder: Design and Implementation

Saeid Gorgin

Leilee Miremoghtadaei

Department of Electrical Engineering and Information Technology, Iranian Research Organization for Science and Technology, Tehran, Iran

ABSTRACT

In the all computer systems, the add operation is an essential part of all computations and operational processes, hence an overall performance efficiency is achievable by increasing the speed of addition. In conventional number systems, the carry propagation causes a dependency between execution time and the operands' length. This issue can be problematic in high-precision operations and it imposes a long delay.

However, by use unconventional redundant number systems, carry free addition is possible. In redundant number system, add operation can be done in constant time. In this paper, according to the importance of decimal computing systems, a new addition algorithm is proposed for redundant decimal numbers. In the suggested algorithm, by using weighted bits set representation and an effective partitioning, the power dissipation and area consumption are decreased without delay penalty.

Keywords: Decimal Addition, Redundant Representation, Signed Digit, Carry Propagation.

Practical Detection of Click Spams Using Classification-Based Algorithms

Mahdieh Fallah

Sajjad Zarifzadeh

Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran

ABSTRACT

Most of today's internet services utilize user feedback (clicks) to improve the quality of their services. For example, search engines use click information as a key factor in document ranking. As a result, some websites cheat to get a higher rank by fraudulently absorbing clicks to their pages. This phenomenon, known as "Click Spam", is initiated by programs called "Click Bot". Thus, the problem of distinguishing bot-generated traffic from the user traffic is critical for the viability of Internet services, like search engines. In this paper, we propose a novel classification-based system to effectively identify fraudulent clicks in a practical manner. We first model user sessions as a set of features. Then, we classify user sessions with a one-class classification algorithm that works based on the well-known K-Nearest Neighbor algorithm. Finally, we analyze our methods with the real log of a Persian search engine. Experimental results show that the proposed algorithm can detect fraudulent clicks with a precision of up to 96% which outperform the previous works by more than 5%.

Keywords: Click Spam, Bot Detection, Anomaly, Machine Learning.

Increasing Throughput in Adhoc Networks Using Queuing Model with Jockeying and Considering Initial Probability of Allocations

Mohammad Amin Keshtkar¹ Ahmad Khonsari^{1,2} Abolfazl Diyanat¹

¹School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran

²School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

ABSTRACT

Wireless ad hoc networks have been considered by researchers for many years. These networks have been used in industry and recently have been employed in car industry namely, Vehicular Ad hoc Networks (VANETs). In this paper, we model delay in VANETs with multiple wireless access technologies in each wireless device embedded in each car. In the proposed model, we employed Jockeying in queuing theory that a customer switches between different queues in order to reduce its waiting time. Thus, jockeying models the switching between different technologies in a wireless device to improve the total latency of an ongoing communication. The results of analytical model have been verified by conducting simulation experiments.

Keywords: Wireless Adhoc Network, VANET, Queuing Model with Jockeying, Multi-Radio.

Human Face Recognition in Digital Images

Alireza Zarei

Aminollah Mahabadi

Department of Electrical Engineering, Shahed University, Tehran, Iran

ABSTRACT

We present a light rapid and accurate method for detecting human faces in color images to constraint on the same nationality. The proposed learning method is different from the classical methods and developed via simple and speed learning to apply on accurate detection of human nationality. The method by using skin model information in color images can decrease the time of authentication without any dependency on lighting, background color, face status, face rotation, image dimension, resolution and imaging condition. The face recognition precision is 98% and 97%rate under the controlled and uncontrolled conditions, respectively. Our suggesting method can be applied as an accurate and rapid light automatic face recognition algorithm in an authentication system.

Keywords: Image Processing, Human Face Recognition, Skin Color, Skin Model.

A Solution Recommender for Exceptions in an Integrated Development Environment

Vahid Amintabar

Abbas Heydarnoori

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT

Exceptions are an indispensable part of a software development process. Developers usually rely on imprecise results from a web search to resolve the exceptions. More specifically, they manually design a query based on the information indicated by the exception message and the stack trace of that exception. Afterwards, they choose and adapt a solution from the web search results. In this scenario, there is a gap between the development environment and web browsers. Moreover, manual search is very time consuming and a lot of useful information from the code is lost. In this paper, we introduce Exception Tracer, an Eclipse plug in that helps developers find solutions of exceptions. Exception Tracer uses the information from the stack trace and the code context where that exception has happened to find relevant code snippets from the Source Forge projects. It then ranks retrieved code snippets from Source Forge projects with the help of GROUM graphs. Finally, variable names are replaced with appropriate names from the developer's code and candidate solutions are shown to him. In addition to candidate code snippets from the Source Forge projects, Exception Tracer retrieves relevant discussions from Stack Overflow. The performance of the proposed approach was measured on a number of code samples from Apache and Java projects and it was shown that for %75 of exceptions, the correct solution is in the 5 topmost results of Exception Tracer. An empirical study with 14 subjects also indicated that Exception Tracer can increase the performance of developers an average of %38 compared to manual web search.

Keywords: Integrated Development Environment, Recommender Systems, Code Snippets, Stack Overflow, Exception Faults, Eclipse Plugin.

The CSI Journal on Computing Science and Information Technology

Vol. 14

No. 1

2016

ABSTRACTS

- **A Solution Recommender for Exceptions in an Integrated Development Environment** 1
Vahid Amintabar and Abbas Heydarnoori
- **Human Face Recognition in Digital Images** 2
Alireza Zarei and Aminollah Mahabadi
- **Increasing Throughput in Adhoc Networks Using Queuing Model with Jockeying and Considering Initial Probability of Allocations** 3
Mohammad Amin Keshtkar, Ahmad Khonsari and Abolfazl Diyanat
- **Practical Detection of Click Spams Using Classification-Based Algorithms** 4
Mahdieh Fallah and Sajjad Zarifzadeh
- **A Low-Power Redundant Decimal Adder: Design and Implementation** 5
Saeid Gorgin and Leilee Miremoghtadaei
- **Design and Characterization of All Graphene Configurable Logic Block for FPGA Applications** 6
Sayed Ali Seif Kashani and Hossein Karimiyan Alidash

The CSI Journal on Computing Science and Information Technology

A Semiannual Publication of Computer Society of Iran (CSI)

Editor-in-Chief

A. Khonsari, Associate Professor, University of Tehran, Tehran, Iran.

Editorial Board

- H. R. Rabiee, Professor, Sharif University of Technology, Iran
H. Sarbazi-azad, Professor, Sharif University of Technology, Iran
K. Faez, Professor, Amirkabir University of Technology, Iran
A. Ghaffarpour Rahbar, Professor, Sahand University of Technology
E. Kabir, Professor, Tarbiat Modares University, Iran
K. Navi, Professor, Shahid Beheshti University, Iran
N. Yazdani, Professor, University of Tehran, Iran
M. H. Yaghmaee Moghaddam, Professor, Ferdowsi University of Mashhad, Iran
M. Analoui, Associate Professor, Iran University of Science & Technology, Iran
M. Ebrahimi Moghaddam, Associate Professor, Shahid Beheshti University, Iran
H. Asadi, Associate Professor, Sharif University of Technology, Iran
A. Akbari, Associate Professor, Iran University of Science & Technology, Iran
R. Berangi, Associate Professor, Iran University of Science & Technology, Iran
H. Pedram, Associate Professor, Amirkabir University of Technology, Iran
N. Moghadam Charkari, Associate Professor, Tarbiat Modares University, Iran
- G. Jaberipur, Associate Professor, Shahid Beheshti University, Iran
J. Habibi, Associate Professor, Sharif University of Technology, Iran
A. H. Jahangir, Associate Professor, Sharif University of Technology, Iran
S. Hesabi, Associate Professor, Sharif University of Technology, Iran
S. H. H. S. Javadi, Associate Professor, Shahed University, Iran
M. Rahgozar, Associate Professor, University of Tehran, Iran
M. Sedighi, Associate Professor, Amirkabir University of Technology, Iran
H. Faili, Associate Professor, University of Tehran, Iran
A. Ghasemi, Associate Professor, K.N. Toosi University of Technology, Iran
M. Abbaspour, Associate Professor, Shahid Beheshti University, Iran
M. Abdollahi Azgomi, Associate Professor, Iran University of Science & Technology, Iran
M. Kargahi, Associate Professor, University of Tehran, Iran
M. Goudarzi, Associate Professor, Sharif University of Technology, Iran
N. Mozayani, Associate Professor, Iran University of Science & Technology, Iran

Assistants

- L. Nourani, Publication Assistant
M. Dolati, Editorial Assistant

Disclaimer: Publication of papers in CSI-CSIT does not imply that the editorial board, reviewers, or CSI-CSIT accept, approve or endorse the data and conclusions of authors.