دانشکده فنی و مهندسی

رشته مهندسی کامپیوتر- گرایش هوش مصنوعی

پایان نامه

جهت اخذ درجه کارشناسی ارشد

تولید خودکار موارد آزمون برای نرم‌افزارهای کاربردی تحت وب

استاد راهنما: دکتر محمدرضا کیوانپور

هاجر همایونی

اسفند ۱۳۹۱

بسم الله الرحمن الرحيم

کلیه دستاوردهای این تحقیق متعلق به دانشگاه الزهرا (س) می‌باشد.

سپاس و ستایش خداوند را که به من توفیق داد تا به یاری و مدد او این پایان نامه را به پایان برسانم.

با تشکر از

استاد راهنمای بزرگوار و دلسوزم جناب آقای دکتر کیوانپور که با دانش و دقت نظر خاص خود همواره مرا در تکمیل این پژوهش راهنمایی کردند،

همسر عزیزم که در تمامی مراحل دلسوزانه پشتیبانم بوده است،

و تمامی کسانی که در طول دوران تحصیل با راهنمایی‌های ارزشمند علمی و معنوی خود، مرا یاری نمودند.

همچنین جا دارد از مؤسسه تحقیقات ارتباطات و فناوری اطلاعات ایران که از این پایان نامه تحت قرارداد شماره ۵۰۰/۱۹۲۳۶/ت به تاریخ ۹۰/۱۲/۲۸ حمایت مادی و معنوی کرده است، قدردانی شود.

چکیده

استفاده از وب یکی از مهمترین، اجتناب ناپذیرترین و در عین حال اقتصادی ترین روش‌ها برای اطلاع رسانی، تجارت و آموزش می‌باشد. همزمان با توسعه وب، تولید نرم‌افزارهای کاربردی تحت وب نسبت به نرم‌افزارهای کاربردی تحت ویندوز اهمیت بیشتری پیدا کرده است. با توجه به گسترش روز افزون کاربرد این نرم‌افزارها، بررسی کیفیت این نرم‌افزارها مسئله‌ای مهم و اساسی تلقی می‌شود. یکی از روش‌های مؤثر برای سنجش کیفیت نرم‌افزار، آزمون نظام‌مند نرم‌افزار است. این آزمون که در طول چرخه حیات نرم‌افزار به صورت مکرر انجام می‌شود، عملیاتی پر هزینه است. تولید موارد آزمون گامی مهم برای آزمون نرم‌افزار محسوب می‌شود. مجموعه این موارد آزمون باید قادر به شبیه سازی کامل و درست اطلاعات ورودی و شرایط کاری واقعی نرم‌افزار باشد. بر این اساس تولید موارد آزمون با توجه به ضرورت و جامعیت و درستی این موارد، فرآیندی پیچیده و پرهزینه است. یکی از مؤثرترین رویکردها برای کاهش هزینه تولید موارد آزمون، خودکارسازی این عملیات است. با وجود تحقیقات زیادی که در زمینه تولید خودکار موارد آزمون نرم‌افزارهای کاربردی تحت وب صورت پذیرفته، این مسئله کماکان دارای چالش-هایی می‌باشد. مهم‌ترین چالش موجود در این زمینه، عدم ارائه مدل توصیفی کامل از نرم‌افزار کاربردی تحت وب جهت تولید موارد آزمون توسط رویکردهای موجود می‌باشد. مدل‌های ارائه شده تا کنون قادر به بیان کلیه رفتارهای نرم‌افزار نیستند. از این رو از طریق آن‌ها نمی‌توان موارد آزمون کارا با قابلیت پوشش بالای نرم‌افزار را تولید نمود. در این تحقیق از یک روش یادگیری نوین مبتنی بر شبکه‌های عصبی فازی جهت استخراج مدل رفتاری کامل به شکل مجموعه‌ای از قواعد فازی از نسل جدید نرم‌افزارهای تحت وب استفاده شده است. از این مدل جهت تولید موارد آزمون کارا با پوشش بالای نرم‌افزار استفاده شده است. مجموعه موارد آزمون تولید شده از طریق روش پیشنهادی حاصل از ترکیب الگوریتم‌های ژنتیک و خنک سازی تدریجی بهینه سازی می‌شوند. مجموعه موارد آزمون تولید شده بر روی دو نرم‌افزار کاربردی تحت وب ۲ استاندارد با عنوان‌های TUDUList و BlindTextGenerator مورد ارزیابی قرار گرفته و با دو روش استاندارد ATUSA و US+CR مقایسه می‌شوند. ارزیابی‌های صورت پذیرفته میزان کارایی ۹۰٪، سطح خودکارسازی ۹۷٪ و میزان پوشش کد ۸۰٪ را برای روش پیشنهادی نشان می-دهد. این نتایج در مجموع نشان‌دهنده عملکرد بهتر روش پیشنهادی نسبت به دو روش مورد بررسی با وجود تعداد موارد آزمون کمتر می‌باشد.

کلمات کلیدی: تولید خودکار موارد آزمون، نرم‌افزارهای تحت وب، شبکه عصبی فازی خودکار مبتنی بر جمعیت

فهرست مطالب

فصل ۱: مقدمه ۱

۱-۱- یادگیری ماشین ۲

۱-۲- طرح مسئله ۳

۱-۳- اهداف و نوآوری‌ها ۶

۱-۴- ساختار پایان نامه ۷

۱-۵- جمع‌بندی ۸

فصل ۲: پیشینه پژوهش ۹

۲-۱- مفاهیم کلیدی ۱۰

۲-۲- معماری کلی سیستم تولید خودکار موارد آزمون نرم‌افزارهای کاربردی تحت وب ۱۴

۲-۳- لزوم تولید خودکار موارد آزمون برای نرم‌افزارهای کاربردی تحت وب ۱۶

۲-۴- چالش‌های تولید خودکار موارد آزمون برای نرم‌افزارهای کاربردی تحت وب ۱۸

۲-۵- رویکردهای تولید خودکار موارد آزمون نرم افزارهای کاربردی تحت وب ۱۹

۲-۵-۱- چارچوب پیشنهادی جهت دسته بندی رویکردهای تولید خودکار موارد آزمون نسل قبلی نرم افزارهای کاربردی تحت وب ۲۰

۲-۵-۱-۱- روش تصادفی ۲۰

۲-۵-۱-۲- روش جعبه سفید ۲۱

۲-۵-۱-۳- روش جعبه سیاه ۲۲

۲-۵-۱-۴- روش مبتنی بر نشست ۲۳

۲-۵-۲- چارچوب پیشنهادی جهت دسته بندی رویکردهای تولید خودکار موارد آزمون نسل جدید نرم افزارهای کاربردی تحت وب ۲۶

۲-۵-۲-۱- روش تصادفی ۲۷

۲-۵-۲-۲- روش مبتنی بر مدل ۲۷

۲-۵-۲-۳- روش مبتنی بر واسط گرافیکی کاربر (GUI) ۳۰

۲-۶- جمع‌بندی ۳۳

فصل ۳: روش پیشنهادی PAFUNN جهت استخراج قواعد فازی ۳۵

۳-۱- شبکه‌های عصبی فازی تکاملی (EFuNN) ۳۶

۳-۱-۱- ویژگی‌های مطلوب شبکه عصبی فازی تکاملی ۳۹

۳-۱-۲- چالش‌های شبکه‌های عصبی فازی تکاملی ۳۹

۳-۲- الگوریتم پیشنهادی PAFuNN جهت استخراج مجموعه قواعد ۴۰

۳-۳- پیاده سازی و آزمون ۴۴

۳-۳-۱- مجموعه داده ۴۶

۳-۳-۲- روش آزمون ۴۷

۳-۳-۳- معیارهای ارزیابی ۴۷

۳-۳-۴- روش‌هایی برای مقایسه با روش پیشنهادی ۴۹

۳-۳-۵- تحلیل نتایج ۵۰
۳-۴- جمع‌بندی ۶۳

فصل ۴: روش RTCGW جهت تولید خودکار موارد آزمون نرم افزارهای کاربردی تحت وب ۶۵

۴-۱- معماری کلی روش تولید خودکار موارد آزمون مبتنی بر قاعده ۶۷
۴-۲- زیرسیستم ایجاد مدل ۶۸
۴-۲-۱- مؤلفه اول: پیمایش تصادفی نرم افزار تحت آزمون ۶۹
۴-۲-۲- مؤلفه دوم: تولید مجموعه قواعد ۷۲
۴-۲-۲-۱- استفاده از PAFUNN جهت استخراج مجموع قواعد معرف رفتار نرم‌افزار تحت وب ۷۳
۴-۳- زیرسیستم تولید موارد آزمون ۷۳
۴-۳-۱- مؤلفه اول: مقداردهی اولیه به مجموعه آزمون ۷۴
۴-۳-۲- مؤلفه دوم: بهینه سازی مجموعه‌های آزمون ۷۶
۴-۳-۲-۱- استفاده از الگوریتم ژنتیک جهت یافتن مجموعه آزمون بهینه ۷۷
۴-۳-۲-۲- استفاده از الگوریتم تقلیدی جهت یافتن مجموعه آزمون بهینه ۸۳
۴-۳-۲-۳- استفاده از الگوریتم تپه‌نوردی جهت یافتن مجموعه آزمون بهینه ۸۶
۴-۳-۲-۴- استفاده از الگوریتم خنک سازی تدریجی جهت یافتن مجموعه آزمون بهینه ۸۷
۴-۳-۲-۵- استفاده از ترکیب الگوریتم‌های ژنتیک و خنک سازی تدریجی جهت یافتن مجموعه آزمون بهینه ۹۰
۴-۴- پیاده سازی و آزمون ۹۳
۴-۴-۱- پیاده سازی زیر سیستم ایجاد مدل ۹۳
۴-۴-۱-۱- پیمایش تصادفی نرم افزار تحت آزمون ۹۳
۴-۴-۱-۲- تولید مجموعه قواعد ۹۵
۴-۴-۲- پیاده سازی زیر سیستم تولید موارد آزمون ۹۵
۴-۴-۲-۱- مقداردهی اولیه به مجموعه آزمون ۹۵
۴-۴-۲-۲- بهینه سازی مجموعه‌های آزمون ۹۶
۴-۴-۳- آزمون ۹۷
۴-۴-۳-۱- نرم‌افزار مورد بررسی ۹۷
۴-۴-۳-۲- روش آزمون ۱۰۰
۴-۴-۴- معیار ارزیابی ۱۰۰
۴-۴-۳-۳- روش‌هایی جهت مقایسه با روش پیشنهادی ۱۰۲
۴-۴-۳-۴- مراحل اجرا برای رسیدن به نتایج و تحلیل نتایج ۱۰۳
۴-۵- جمع‌بندی ۱۱۵

فصل ۵: نتیجه گیری و توسعه‌های آتی ۱۱۶

۵-۱- نتیجه‌گیری ۱۱۷
۵-۲- توسعه‌های آتی ۱۱۸

فهرست جداول و اشکال

جدول ۲-۱- رویکردهای موجود برای تولید خودکار موارد آزمون نسل قدیمی نرم‌افزارهای تحت وب ۲۵

جدول ۲-۲- ارزیابی رویکردهای تولید خودکار موارد آزمون نرم‌افزارهای تحت وب نسل قبل ۲۶

جدول ۲-۳- رویکردهای تولید خودکار موارد آزمون نسل جدید نرم‌افزارهای تحت وب ۳۲

جدول ۲-۴- ارزیابی رویکردهای تولید خودکار موارد آزمون نرم‌افزارهای تحت وب نسل جدید ۳۳

جدول۳-۱- مقایسه شبکه‌های عصبی EFuNN، sEFuNN، rEFuNN، pEFuNN، aEFuNN و PAFuNN بر روی مجموعه داده Furnace ۵۱

جدول ۳-۲- تاثیر انتخاب ماشین‌های یادگیر متفاوت در نتایج حاصل از شبکه PAFuNN بر روی مجموعه داده Furnace ۵۳

جدول ۳-۳- مقایسه شبکه‌های عصبی EFuNN، sEFuNN، rEFuNN، pEFuNN، aEFuNN و PAFuNN بر روی مجموعه داده Iris ۵۸

جدول ۳-۴- تاثیر انتخاب ماشین‌های یادگیر متفاوت در نتایج حاصل از شبکه PAFuNN بر روی مجموعه داده Iris ۵۹

جدول ۴-۱- شرح اختصارات فصل ۶۷

جدول ۴-۲- مثال‌هایی از عناصر و رویدادها ۷۰

جدول ۴-۳- بردار ورودی- خروجی حاصل از پیمایش تصادفی نرم افزار تحت وب ۷۲

جدول ۴-۴- انواع مؤلفه‌ها، انواع رویدادها، انواع ورودی‌ها و کد مربوط به هر یک برای ذخیره سازی در بردار ویژگی‌ها ۷۲

جدول ۴-۵- نمونه‌ای از بردار ورودی- خروجی حاصل از پیمایش تصادفی نرم افزار تحت وب ۷۲

جدول ۴-۶- ساختار یک مورد آزمون ۷۴

جدول۴-۷- مقادیر تنظیم شده برای پارامترهای الگوریتم ژنتیک ۹۷

جدول ۴-۸- میانگین مربع خطا، تعداد نود قانون تولید شده و زمان CPU صرف شده حاصل از آموزش شبکه PAFUNN بر روی داده‌های حاصل از پیمایش تصادفی نرم‌افزار TUDU ۱۰۴

جدول ۴-۹- میزان برازش و زمان صرف شده توسط الگوریتم‌های بهینه سازی مجموعه موارد آزمون ۱۰۵

جدول ۴-۹- معیارهای اندازه گیری شده زمانی که الگوریتم‌های بهینه سازی مجموعه موارد آزمون مطابق با جدول قرار گرفته باشند ۱۰۷

جدول ۴-۱۱- مقایسه روشهای پیشنهادی، US، CR و US+CR بر اساس معیارهای کارایی، پوشش کد و سطح اتوماسیون ۱۰۸

جدول ۴-۱۲- میانگین مربع خطا، تعداد نود قانون تولید شده و زمان CPU صرف شده حاصل از آموزش شبکه PAFUNN بر روی داده‌های حاصل از پیمایش تصادفی نرم‌افزار Blind Text Generator ۱۱۰

جدول ۴-۱۳- میزان برازش و زمان صرف شده توسط الگوریتم‌های بهینه سازی مجموعه موارد آزمون ۱۱۱

جدول ۴-۱۴- معیارهای اندازه گیری شده زمانی که الگوریتم‌های بهینه سازی مجموعه موارد آزمون مطابق با جدول قرار گرفته باشند ۱۱۳

جدول ۴-۱۵- مقایسه روش پیشنهادی با روش ATUSA بر اساس معیارهای کارایی، پوشش کد و سطح اتوماسیون ۱۱۴

شکل ۲-۱- (الف) معماری نسل قدیم و (ب) جدید نرم‌افزارهای کاربردی تحت وب ۱۳

شکل ۲-۲- نقش اوراکل آزمون ۱۴

شکل ۲-۳- چارچوب پیشنهادی جهت دسته بندی رویکردهای موجود برای تولید خودکار موارد آزمون نسل قدیمی نرم‌افزارهای کاربردی تحت وب ۱۵

شکل ۲-۴- چارچوب پیشنهادی جهت دسته بندی رویکردهای موجود برای تولید خودکار موارد آزمون نسل قدیمی نرم‌افزارهای کاربردی تحت وب ۲۰

شکل ۲-۵- معماری مبتنی بر نشست تولید موارد آزمون ۲۴

شکل ۲-۶- چارچوب پیشنهادی جهت دسته بندی رویکردهای تولید خودکار موارد آزمون برای نسل جدید نرم‌افزارهای کاربردی تحت وب ۲۷

شکل ۲-۷- معماری خزنده نرم‌افزار تحت وب با عنوان CRAWLAJAX ۲۹

شکل۲-۸- ساختار کلی سیستم تولید خودکار موارد آزمون ۳۳

شکل ۳-۱- فلوچارت روش EFUNN ۳۸

شکل ۳-۲- ساختار کلی PAFUNN ۴۱

شکل ۳-۳- فلوچارت روش پیشنهادی PAFUNN ۴۲

شکل ۳-۴- اتصال داخلی ماشین یادگیر به شبکه PAFUNN ۴۴

شکل ۳-۵- الگوریتم EFUNN ۴۵

شکل ۳-۶- شبه کد PAFUNN برای استخراج قواعد ۴۶

شکل ۳-۷- مقایسه شبکه‌های عصبی EFuNN، sEFuNN، rEFuNN، pEFuNN، aEFuNN و PAFuNN بر روی مجموعه داده Furnace ۵۲

شکل۳-۸- تاثیر انتخاب ماشین‌های یادگیر متفاوت در نتایج حاصل از شبکه PAfuNN بر روی مجموعه داده Furnace ۵۳

شکل ۳-۹- فرآیند تکامل شبکه‌های EFuNN، sEFuNN، rEFuNN، pEFuNN، aEFuNN و PAEFuNN بر روی مجموعه داده furnace. در ستون اول مقادیر پیش بینی شده و واقعی داده‌ها با یک گام تاخیر زمانی که شبکه بر روی نیمه اول داده‌ها مورد آزمون قرار می‌گیرد. در ستون دوم شبکه آموزش دیده شده بر روی نیمه دوم داده‌ها مورد آزمون قرار می‌گیرد. ۵۶

شکل ۳-۱۰- مقایسه شبکه‌های عصبی EFuNN، sEFuNN، rEFuNN، pEFuNN، aEFuNN و PAFuNN بر روی ۵۸

مجموعه داده Iris

شکل۳-۱۱- تاثیر انتخاب ماشین‌های یادگیر متفاوت در نتایج حاصل از شبکه PAFuNN بر روی مجموعه داده Iris ۶۰

شکل ۳-۱۲- فرآیند تکامل شبکه‌های EFuNN، sEFuNN، rEFuNN، pEFuNN، aEFuNN و PAEFuNN بر روی مجموعه داده Iris. در ستون اول مقادیر پیش بینی شده و واقعی داده‌ها با یک گام تاخیر زمانی که شبکه بر روی نیمه اول داده‌ها مورد آزمون قرار می‌گیرد. در ستون دوم شبکه آموزش دیده شده بر روی نیمه دوم داده‌ها مورد آزمون قرار می‌گیرد. ۶۳

شکل ۴-۱- معماری کلی روش پیشنهادی ۶۸

شکل ۴-۲- ساختار کلی سیستم تولید خودکار موارد آزمون ۶۸

شکل ۴-۳- معماری کلی زیرسیستم ایجاد مدل ۶۹

شکل ۴-۴- فلوچارت کلی پیمایش تصادفی نرم افزار تحت وب ۷۱

شکل ۴-۵- معماری پیشنهادی برای تولید موارد آزمون ۷۴

شکل ۴-۶- مقداردهی اولیه به مجموعه آزمون از طریق پیمایش شبه تصادفی نرم‌افزار ۷۵

شکل ۴-۷- (الف) نمونه‌ای از یک مجموعه آزمون (ب) گذار حالت برای این مجموعه آزمون ۷۶

شکل ۴-۸- فلوچارت الگوریتم ژنتیک جهت بهینه سازی مجموعه موارد آزمون ۷۸

شکل ۴-۹- تلفیق عمودی ۸۱

شکل ۴-۱۰- تلفیق افقی ۸۲

شکل۴-۱۱- الگوریتم تقلیدی (سیاست اول) ۸۴

شکل ۴-۱۲- الگوریتم تقلیدی (سیاست دوم) ۸۵

شکل ۴-۱۳- فلوچارت الگوریتم تپه‌نوردی ۸۷

شکل ۴-۱۴ فلوچارت الگوریتم خنک سازی تدریجی ۸۹

شکل۴-۱۵- ترکیب الگوریتم ژنتیک و خنک سازی تدریجی (سیاست اول) ۹۱

شکل۴-۱۶- ترکیب الگوریتم ژنتیک و خنک سازی تدریجی (سیاست دوم) ۹۲

شکل ۴-۱۷- شبه کد پیمایش تصادفی نرم افزار تحت وب ۹۴

شکل ۴-۱۸- نمونه‌ای از قواعد فازی حاصل از اجرای PAFUNN بر روی مجموعه داده‌های حاصل از مرحله پیمایش تصادفی نرم-افزار تحت وب ۹۵

شکل۴-۱۹- مقدار دهی اولیه به مجموعه آزمون ۹۶

شکل ۴-۲۲- شمای کلی نرم افزار TUDU ۹۸
شکل ۴-۲۳- شمای کلی نرم افزار Blind Text Generator ۹۹
شکل ۴-۲۴- معیارهای موثر بر کارایی مجموعه آزمون ۱۰۰
شکل ۴-۲۵- فرآیند تکامل و آموزش شبکه PAFUNN بر روی داده‌های حاصل از پیمایش تصادفی نرم‌افزار TUDU ۱۰۳
شکل ۴-۲۶- نمونه‌هایی از قواعد تولید شده حاصل از آموزش شبکه PAFUNN ۱۰۴
شکل ۴-۲۷- میزان برازش و زمان صرف شده توسط الگوریتم‌های بهینه سازی مجموعه موارد آزمون ۱۰۶
شکل ۴-۲۸- معیارهای اندازه گیری شده زمانی که الگوریتم‌های بهینه سازی مجموعه موارد آزمون مطابق با شکل قرار گرفته باشند ۱۰۷
شکل ۴-۲۹- مقایسه روشهای پیشنهادی، US، CR و US+CR بر اساس معیارهای کارایی، پوشش کد و سطح اتوماسیون ۱۰۸
شکل ۴-۳۰- فرآیند تکامل و آموزش شبکه PAFUNN بر روی داده‌های حاصل از پیمایش تصادفی نرم‌افزار Blind Text Generator ۱۱۰
شکل ۴-۳۱- نمونه‌هایی از قواعد تولید شده حاصل از آموزش شبکه PAFUNN ۱۱۰
شکل ۴-۳۲- میزان برازش و زمان صرف شده توسط الگوریتم‌های بهینه سازی مجموعه موارد آزمون ۱۱۲
شکل ۴-۳۳- معیارهای اندازه گیری شده زمانی که الگوریتم‌های بهینه سازی مجموعه موارد آزمون مطابق با جدول قرار گرفته باشند ۱۱۳
شکل ۴-۳۴- مقایسه روش پیشنهادی با روش ATUSA بر اساس معیارهای کارایی، پوشش کد و سطح اتوماسیون ۱۱۴

فصل ۱: **مقدمه**

وب که عامل اصلی ساختار اینترنت میباشد اکنون بدلیل کاربرد مفید و کاربران بسیار زیاد آن به یکی از مؤثرترین روش‌های ارتباطی تبدیل شده و همین امر موجب نیاز روز افزون به برنامه‌های کاربردی تحت وب شده است. با توجه به گسترش روز افزون کاربردهای این نرم‌افزارها، بررسی کیفیت این نرم‌افزارها از اهمیت ویژه‌ای برخوردار است. یکی از ابزارها برای بررسی کیفیت نرم‌افزارها آزمون نرم‌افزاری است. آزمون دستی این نرم‌افزارها نیازمند تلاش زیاد و هزینه‌های بسیار بالا است. یکی از مهم ترین وظایف در آزمون نرم‌افزار تولید موارد آزمون است . در این تحقیق تولید موارد آزمون نرم‌افزاری برای نرم‌افزارهای کاربردی تحت وب با بهره گیری از یک روش یادگیری ماشین نوین خودکارسازی می‌شوند. بنابراین موجب تسهیل وظیفه آزمون و در نهایت افزایش کیفیت نرم‌افزار تحت وب می‌شود.

۱-۱- یادگیری ماشین

در انسان‌ها، یادگیری از طریق فرآیندهای مشخص و تغییرات در سیستم عصبی اتفاق می‌افتد. فعالیت یادگیری معمولا شامل مجموعه‌ای از مشاهدات در رابطه با پدیده‌های طبیعی و فرآیندی برای تبدیل آن اطلاعات به دانش می‌باشد.

سیستم‌های مصنوعی از طریق تغییراتی که آن‌ها به نمایش‌های داخلی خودشان، داده‌ها، مدل‌ها و/یا ساختارها (در سطح نرم‌افزاری و سخت افزاری) می‌دهند، یادگیری می‌کنند. این ماشین‌ها برای حل وظایف معینی ساخته می‌شوند و هدف از یادگیری آن‌ها، بهبود کارایی‌شان در آن وظایف توسط یادگیری از محیط، آموزگاران، و تجربیات خودشان در دستیابی به اهداف مشخص می‌باشد. روش‌های یادگیری ماشین برای وظایفی که پیچیدگی مسئله از لحاظ ورودی‌ها و خروجی‌های متفاوت بسیار زیاد است که برنامه‌نویس نمی‌تواند آن‌ها را به صورت صریح کد کند. در یک دسته بندی کلی، رویکردهای موجود در زمینه یادگیری ماشین را می‌توان در سه دسته نظارتی، بدون نظارت و نیمه نظارتی تقسیم نمود [1].

- یادگیری نظارتی[1]:

روش‌های یادگیری ماشین معمولا از طریق نیاز آن‌ها به نوع داده‌ای که می‌توانند استفاده کنند طبقه بندی می‌شوند. الگوریتم‌هایی با نظارت کامل، همیشه به یک مربی برای آشکار کردن برچسب داده‌ها نیاز دارند. هدف این الگوریتم‌ها یادگیری نگاشتی از داده‌های ورودی به خروجی مشخص شده توسط مربی می‌باشد.

[1] Supervised learning

- یادگیری بدون نظارت[1]:

یادگیری بدون نظارت، فقط با داده‌های آموزشی و بدون نیاز به مربی کار می‌کنند. آن‌ها معمولا سعی می-کنند که الگوی شباهتی میان ورودی‌هایشان پیدا کنند و خوشه‌هایی از نمونه‌های داده‌های آموزشی را تشخیص دهند.

- یادگیری نیمه نظارتی[2]:

نوعی از یادگیری است که با استفاده از مثال‌های آموزشی برچسب دار و بدون برچسب برای پیش بینی برچسب مثال جدید انجام می‌شود. در بسیاری از دامنه‌های یادگیری عملی (مانند طبقه‌بندی متن و تصویر و زیست فناوری)، داده‌های بدون برچسب زیادی به همراه داده‌های برچسب دار محدود وجود دارد. همچنین در بیشتر موارد تولید داده‌های برچسب دار هزینه زیادی دارد.

در این پژوهش یک روش یادگیری نظارتی/ بدون نظارت پیشنهاد و از آن جهت تولید خودکار موارد آزمون نرم‌افزارهای کاربردی تحت وب استفاده شده است.

۱-۲- طرح مسئله

رایانه‌ها و برنامه‌ها در بسیاری از جنبه‌های زندگی ما گسترش یافته‌اند، نظیر معاملات الکترونیکی، رسانه‌ها، حمل و نقل، پزشکی و سلامت. قابلیت اعتماد نرم‌افزار و سخت افزار، موضوعات بسیار مهمی هستند زیرا بسیاری از جنبه-های فعالیت‌های روزانه ما و یا حتی زندگی ما بستگی به آن‌ها دارد. آزمون نرم‌افزار راهی‌است برای وارسی درستی و تناسب یک سیستم نرم‌افزاری، و یا ، به طور جایگزین، برای حصول اطمینان از اینکه یک سیستم نرم‌افزاری نیازهای مطرح شده در توصیف را برآورده می‌کند. در حالی که آزمون نرم‌افزاری بسیار حائز اهمیت است، بسیار پر هزینه است زیرا می‌بایست در طول چرخه حیات نرم‌افزار صورت پذیرد. زمان و تلاشی که برای آزمون نرم‌افزار صرف می‌شود معمولا بیشتر از کل هزینه پیاده سازی نرم‌افزار است [2]. در صورتی که فرآیند آزمون نرم‌افزار خودکارسازی شود، می‌تواند به طور مؤثر باعث کاهش هزینه توسعه نرم‌افزار شود [3]. به طور کلی آزمون نرم‌افزار شامل سه مرحله می‌باشد: تولید موارد آزمون، اجرای آزمون و ارزیابی آزمون. تولید موارد آزمون برای موفقیت آزمون بخش حیاتی به شمار می‌آید، زیرا آزمون کامل برنامه داده شده غیر ممکن است چرا که تعداد موارد آزمون

[1] Unsupervised learning
[2] Semi-Supervised learning

برای آزمون کامل یک برنامه بینهایت است، و یک طراحی مناسب موارد آزمون نیاز است تا تعداد زیادی از خطاها را کشف کند [4]. تا به امروز تولید موارد آزمون یک فرآیند دستی باقی مانده است و خودکارسازی‌های کمی در این زمینه صورت گرفته است. تولید موارد آزمون، شاید هزینه بر ترین زیرپروسه در آزمون نرم‌افزار باشد (حدود ۴۰ درصد از کل هزینه آزمون نرم‌افزار). بنابراین هزینه کل آزمون نرم‌افزار می‌تواند به شدت کاهش یابد اگر این فرایند خودکارسازی شود [5].

انتشار گسترده اینترنت، تقاضای روزافزون کاربردهای تحت وب را همراه با نیازهایی نظیر قابلیت اطمینان، قابلیت استفاده، قابلیت همکاری و امنیت بالا، موجب شده است. با توجه به فشار بازار و زمان محدود معمولا از آزمودن نرم‌افزارهای تحت وب صرف نظر می‌شود، زیرا فرآیندی زمانبر و پرهزینه است [6]. این کار بر کیفیت نرم-افزارهای تحت وب تاثیر منفی می‌گذارد، بنابراین موجب به وجود آمدن نیاز برای روش‌های آزمون کارا و کم هزینه آن‌ها می‌شود.

با وجودی که آزمون کاربردهای تحت وب ویژگی‌های مشترکی با آزمون سیستم‌های قدیمی دارد، در بیشتر مواقع تئوری‌ها و روش‌های آزمون‌های قدیمی نمی‌تواند بدون تغییر بر روی کاربردهای تحت وب اعمال شود، و این به علت پیچیدگی و ویژگی‌های منحصر به فرد کاربردهای تحت وب است.

یک کاربرد تحت وب را می‌توان به صورت یک سیستم توزیع شده، با معماری چندلایه یا سرویس گیرنده-سرویس دهنده در نظر گرفت که شامل ویژگی‌های زیر است:

- تعداد زیادی کاربر که درتمام دنیا توزیع شده‌اند و به صورت همزمان به آن دسترسی دارند.
- محیط‌های اجرای ناهمگن، شامل سخت‌افزارها ، اتصالات شبکه، سیستم‌های عامل، سرویس‌دهنده‌ها و مرورگرهای وب متفاوت
- طبیعت به شدت ناهمگن مبتنی بر گستره‌ای از مؤلفه‌های نرم‌افزاری که می‌توانند توسط تکنولوژی‌های متفاوتی (زبان‌های برنامه نویسی و مدل‌های متفاوت) ایجاد شوند.
- قابلیت تولید مؤلفه‌های نرم‌افزار در زمان اجرا با توجه به ورودی‌های کاربر و وضعیت سرویس دهنده.

هر یک از جنبه‌های مطرح شده در قسمت فوق، چالش‌های جدیدی در زمینه آزمون مطرح می‌کند.

آزمون کاربردهای تحت وب شامل اجرای آن‌ها با استفاده از ترکیبی از ورودی‌ها و حالات به منظور آشکارسازی شکست‌های نرم‌افزار می‌باشد. یک شکست تجلی عدم قابلیت یک سیستم یا مؤلفه برای انجام یک عمل مطرح شده

در نیازمندی‌های توصیف می‌باشد. یک شکست می‌تواند نتیجه هر گونه خطا در پیاده سازی کاربرد باشد. به طور کلی علت شکست‌ها می‌تواند خطاهای موجود در خود نرم‌افزار و یا خطاهای موجب شده توسط محیط اجرا و یا واسط بین نرم‌افزار و محیط اجرا باشد[7] .

همانطور که اشاره شد تولید موارد آزمون در موفقیت آزمون بخش حیاتی به شمار می‌آید. یک مورد آزمون برای یک کاربرد تحت وب نسل قبل را می‌توان به صورت ترتیبی از صفحات برای مشاهده، به علاوه مقادیر ورودی به صفحات تعریف نمود.

با روی کار آمدن نسل جدید وب، و استفاده از تکنولوژی‌های جدید برای تولید کاربردهای تحت وب (نظیر AJAX[1])، روش‌های قبلی آزمون مبتنی بر چند-صفحه قابل اعمال نبوده و چالش‌های جدیدی در زمینه تولید خودکار موارد آزمون برای این کاربردها مطرح شده است. از جمله مهمترین این چالش‌ها می‌توان به موارد زیر اشاره نمود [8].

- ماهیت مبتنی بر رویداد نسل جدید وب، اولین چالش جدی آزمون آن‌ها را موجب می‌شود. زیرا به جای ایجاد و ارسال URL ها به سمت سرویس دهنده، مدل رویدادهای مرورگر می‌بایست ایجاد شود. بنابراین شبیه سازی رویدادهای کاربر بر روی واسط نرم‌افزار نیازمند محیطی است که به تمامی تکنولوژی‌های لازم (نظیر Javascript، XMLHttpRequest، DOM[2]) مجهز شده باشد.
- در نسل جدید نرم‌افزارهای تحت وب هر پاسخ به یک رویداد سمت سرویس دهنده به واسط تک-صفحه‌ای تزریق می‌شود. بنابراین خطاها در سطح DOM انتشار می‌یابند. از این رو دسترسی به DOM پویا در زمان اجرا برای تحلیل و کشف خطاهای انتشار یافته لازم و ضروری است.
- مقادیر مناسب برای مقدار دهی به عناصر ورودی به DOM ها باید انتخاب شوند.
- تعریف مورد آزمون برای این دسته از کاربردها نیز با توجه به ماهیت تک - صفحه‌ای آن‌ها متفاوت از قبل است. یک تعریف می‌تواند به صورت ترتیب عملیات انجام شده توسط کاربر باشد.

با توجه به نوپا بودن این نسل از نرم‌افزارهای تحت وب، تا کنون روش‌های بسیار محدودی برای تولید خودکار موارد آزمون برای این دسته از نرم‌افزارها ارائه شده است که با چالش‌های متعددی روبرو هستند. از جمله این چالش‌ها می‌توان به موارد زیر اشاره نمود [9, 10, 11].

- عدم ارائه یک روش کارا برای دستیابی به میزان پوشش کافی نیازهای آزمون نرم‌افزار

[1] Asynchronous JavaScript and XML
[2] Document Object Model

- نیاز به تلاش‌های دستی زیاد در کنار فرآیند خودکارسازی تولید موارد آزمون نرم‌افزار
- وابستگی روش‌های ارائه شده تا کنون به یک مدل اولیه از نرم‌افزار که همه ویژگی‌های نرم‌افزار را شامل نمی-شود.
- عدم ارائه یک مدل رفتاری کامل توسط روش‌های مطرح شده تا کنون که قادر به پوشش تمامی رفتارهای نرم-افزار تحت وب باشد و بتوان از آن جهت تولید موارد آزمون کارا و کامل استفاده نمود.

تلاش این تحقیق بر بهره گیری از یک روش نوین یادگیری مبتنی بر شبکه‌های عصبی فازی تکاملی به منظور ایجاد یک مدل رفتاری کامل در قالب مجموعه‌ای از قواعد فازی به صورت پویا و برخط و استفاده از آن جهت تولید خودکار موارد آزمون نسل جدید نرم‌افزارهای کاربردی تحت وب به منظور برآورده ساختن نیازهای آزمون آن‌ها می‌باشد.

۱-۳- اهداف و نوآوری‌ها

هدف از انجام این پژوهش، ارائه راه حل برای چالش‌هایی می‌باشد که در بخش طرح مسئله ذکر شد. در ادامه اشاره‌ای به نوآوری‌هایی که در این پژوهش ارائه شده است، خواهد شد. این نوآوری‌ها به سه دسته کلی تقسیم می-شوند: فرآیند مدلسازی نرم‌افزار، تولید موارد آزمون با استفاده از آن مدل، و ارزیابی موارد آزمون تولید شده که نوآوری‌های هر کدام از آن‌ها به صورت جداگانه ذکر خواهد شد.

- فرآیند مدل‌سازی نرم‌افزار
  - تولید مجموعه‌ای از قواعد فازی معرف رفتار نرم‌افزار به منظور تولید موارد آزمون: با توجه به پیچیدگی بالای کاربردهای تحت وب امروزی، داشتن یک مدل رفتاری از این کاربردها در تولید موارد آزمون می‌تواند بسیار مؤثر واقع شود. در تحقیقات به عمل آمده تا کنون اغلب نرم‌افزار را به صورت یک گراف مدل می‌کنند که در آن نودها معرف حالت‌های نرم‌افزار و یال‌ها معادل عوامل انتقال بین حالات (نظیر رویدادها و لینک‌ها) و هزینه مربوط به هر انتقال، و میزان پوشش کسب شده از انجام آن انتقال می‌باشد. با توجه به اینکه مدل‌های مبتنی بر گراف کارایی لازم را دارا نمی‌باشند و نیز با توجه به ناکامل بودن اکثر این مدل‌ها و وابسته بودن آن‌ها به یک نمونه اولیه از نرم‌افزار و عدم ارائه کلیه رفتار سیستم، در این پژوهش به جای استفاده از گراف از یک مجموعه قواعد فازی معرف رفتار سیستم برای مدل‌سازی نرم‌افزار استفاده می‌شود.

- ارائه یک الگوریتم یادگیری نظارتی/ بدون نظارت جدید جهت تولید مجموعه قواعد فازی معرف رفتار نرم‌افزار تحت وب: این الگوریتم که نمونه‌ای از شبکه‌های عصبی فازی را پیاده سازی می‌کند، روشی نو برای ایجاد قاعده از روی مجموعه داده‌های برچسب دار ارائه می‌دهد. بر خلاف روش‌های قبلی که در آن‌ها به ازای هر نمونه داده آزمایش ساختار شبکه تکامل می‌یابد در این روش بر اساس جمعیتی از نمونه‌های ورودی ساختار شبکه تکامل می‌یابد. این مسئله منجر به کاهش پیچیدگی‌های ساختاری در شبکه پیشنهادی می‌شود. در این شبکه همچنین از ماشین‌های یادگیر برای انتخاب حدود آستانه مورد نیاز استفاده شده است. این امر موجب کاهش میزان خطای شبکه می‌شود.

- فرآیند تولید موارد آزمون از روی مدل اولیه
  - ارائه روشی تکاملی مبتنی بر ترکیب الگوریتم‌های تکاملی ژنتیک و خنک سازی تدریجی[1] به منظور تولید مجموعه موارد آزمون بهینه با استفاده از مدل رفتاری مرحله قبل که منتهی به بیشترین میزان پوشش نرم-افزار گردد.
  - ارائه یک معیار پوشش جدید بر مبنای کد درهم ساز[2] حالات مختلف نرم‌افزار.
- فرآیند ارزیابی موارد آزمون تولید شده
  - ارائه یک چارچوب کلی برای دسته بندی معیارهای ارزیابی موارد آزمون: در این چارچوب معیارهای ارزیابی ارائه شده جهت بررسی موارد آزمون در یک دسته بندی کلی مطرح شده است. سپس از این معیارها جهت ارزیابی مورد آزمون تولید شده استفاده شده است.

۱-۴- ساختار پایان نامه

ساختار فصول آتی پایان‌نامه به شرح زیر است:

فصل دو نگاهی کلی به پژوهش‌های صورت پذیرفته و نیز مفاهیم کلیدی مورد نیاز برای درک بهتر از روش پیشنهاد شده خواهد داشت، فصل سوم شامل ارائه روش PAFuNN جهت استخراج قواعد فازی از مجموعه نمونه-های ورودی و فصل چهارم شامل استفاده از این روش در تولید مدل رفتاری از نرم‌افزار تحت وب و در نهایت تولید مجموعه موارد آزمون با استفاده از مدل حاصل می‌باشد. در نهایت فصل پنجم نتیجه‌گیری و توسعه‌های آتی را شامل می‌شود.

[1] Simulated Anealing
[2] Hash code

۱-۵- جمع‌بندی

در این فصل مسئله‌ی مورد بررسی در پژوهش به تفصیل مطرح شد و پس از آن اهداف و نوآوری‌های برجسته‌ی پژوهش با جزئیات مورد بحث و بررسی قرار گرفت. این فصل نگاهی کلی بر اهداف و نوآوری‌های مطرح در این پژوهش ارائه داده است.

فصل ۲: **پیشینه پژوهش**