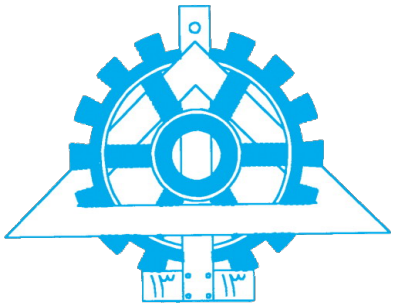


دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر
گروه مهندسی نرم‌افزار

یادگیری تطبیقی مدل خط محصول‌های نرم‌افزار با رویکرد مبتنی بر درخت

پروژه کارشناسی مهندسی کامپیوتر گرایش مهندسی نرم‌افزار

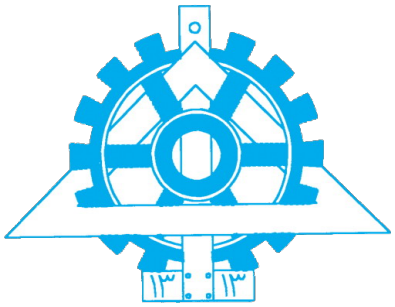
بهار امامی افشار

استاد راهنما

دکتر رامتین خسروی

تیر ۱۴۰۲


بسم الله الرحمن الرحيم



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر
گروه مهندسی نرم‌افزار

یادگیری تطبیقی مدل خط محصول‌های نرم‌افزار با رویکرد مبتنی بر درخت

پروژه کارشناسی مهندسی کامپیوتر گرایش مهندسی نرم‌افزار

بهار امامی افشار

استاد راهنما

دکتر رامتین خسروی

تیر ۱۴۰۲

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر

گواهی دفاع از پروژه کارشناسی

هیأت داوران پروژهٔ کارشناسی آقای / خانم بهار امامی افشار به شمارهٔ دانشجویی ۸۱۰۱۹۷۶۶۲ در رشتهٔ مهندسی کامپیوتر - گرایش مهندسی نرم‌افزار را در تاریخ ..... با عنوان «یادگیری تطبیقی مدل خط محصول‌های نرم‌افزار با رویکرد مبتنی بر درخت»

	به عدد	به حروف
با نمرهٔ نهایی		

و درجهٔ [ ] ارزیابی کرد.

ردیف	مشخصات هیأت داوران	نام و نام خانوادگی	مرتبهٔ دانشگاهی	دانشگاه یا مؤسسه	امضا
۱	استاد راهنما	دکتر رامتین خسروی	استادیار	دانشگاه تهران	
۲	استاد داور داخلی	دکتر فاطمه قاسمی	استادیار	دانشگاه تهران	

نام و نام خانوادگی معاون تحصیلات تکمیلی و پژوهشی دانشکده / گروه:

تاریخ و امضا:

نام و نام خانوادگی معاون آموزشی و تحصیلات تکمیلی پردیس دانشکده‌های فنی:

تاریخ و امضا:

تعهدنامهٔ اصالت اثر

باسمه تعالی

اینجانب بهار امامی افشار تأیید می‌کنم که مطالب مندرج در این پروژه حاصل کار پژوهشی اینجانب است و به دستاوردهای پژوهشی دیگران که در این نوشته از آن‌ها استفاده شده است مطابق مقررات ارجاع گردیده است. این پروژه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتری ارائه نشده است.

نام و نام خانوادگی دانشجو: بهار امامی افشار

تاریخ و امضای دانشجو:

چکیده

یادگیری تطبیقی روشی نوین در راستای تحلیل عملکرد مدل خط محصول‌های نرم‌افزار و آزمون مبتني بر مدل است. در این نوع از یادگیری تمرکز بر روی ایجاد مدل رفتاری خط محصولات می‌باشد. به دلیل نقاط مشترک بین محصولات متفاوت یک خط محصول، امکان استفاده از مدل‌های یادگیری شده قبلي به وجود مي‌آید به نحوی که استفاده از این ویژگي‌ها می‌تواند منجر به بهبود عملکرد یادگیری شود.

مدل‌های رفتاری اغلب در دنیای واقعی وجود ندارند یا از رده خارج شده‌اند و به همین دلیل برای تحلیل بیشتر نیاز دارند که از روی پیاده‌سازی‌های قبلی مجددا ساخته بشوند. در این مواقع ترجیح بر این است که از یادگیری مبتني بر مدل استفاده شود. متغیرهای زیادی باعث ایجاد چالش در این نوع یادگیری مي‌شوند که مهمترین آنها تعداد جستجوهای انجام‌شده می‌باشد که در بسیاری از مواقع یادگیری را غیرممکن می‌سازد. به همین منظور ضرور ی است که الگوریتم های یادگیری تطبیقي به کار روند تا با ذخیره داده‌های جستجوی قبلی باعث بهبود عملکرد یادگیری شوند.

در این پروژه قصد بر این داشتیم تا با به کارگیری مدل تطبیقي بر روی الگوریتم مبتني بر درخت کرنز و وزیرانی، میزان بهبود عملکرد آن را بسنجیم. دو جزو مهم یادگیری مبتنی بر مدل جستجو های عضویت و معادلسازی می‌باشند. معیارهای اصلي سنجش عملکرد در این پژوهش تعداد نوبت‌های یادگیری ، تعداد جستجوهای معادلسازی، تعداد بازنشانی‌ها و تعداد کل نمادهای ورودی استفاده‌شده در یادگیری ،خواهند بود. به این منظور پس از پیاده سازی اولیه الگوریتم غیرتطبیقی، درخت استفاده‌شده در یادگیری را به دست آورده و با استفاده از الگوریتم تعریف شده در این مقاله فایلی از ورودی‌های و خروجی‌های متناسبشان استخراج کردیم. در مرحله بعدی، الگوریتم یادگیری تظبیقی پیاده سازی شد که که با دریافت فایل حاصل از یادگیری محصول پیشین، درختی بر اساس این اطلاعات می سازد و با استفاده از آن محصول نرم‌افزاری جدید را یادگیری می‌کند. به این صورت با استفاده‌ی مجدد از جسنجوهای محصول پیشین تعداد جستجوهای محصولات بعدی کاهش خواهد یافت که گام مهمی در پردازش و بادگیری مبتنی بر مدل محصولات خط نرم‌افزار خواهد بود.

واژگان کلیدی یادگیری تطبیقی مبتنی بر درخت، مدل رفتاری، خط محصول نرم‌افزاری، الگوریتم کرنز و وزیرانی

فهرست مطالب

کتاب‌نامه **اول**

فهرست تصاویر

فهرست جداول

فصل ۱

مقدمه

۱.۱ مقدمه

یادگیری مدل توسعه‌ای خط محصولات نرم‌افزاری به دلیل اهمیت بسیار زیادی که در دامنه‌های متفاوت از جمله خودروسازی، مخابرات، امور مالی و بهداشت و درمان دارند در سال‌های اخیر توجه بسیاری از محققین را به خود معطوف ساخته‌اند. خط محصولات نرم‌افزار از تعدادی از سیستم‌های نرم‌افزار فشرده که مجموعه‌ای از ویژگی‌ها یا عملکردهای مشترک را به اشتراک می‌گذارند، تشکیل می‌شود. استفاده مجدد از ویژگی‌های مشخص در چندین محصول منجر به افزایش کارایی و بهره‌وری در توسعه، بهبود کیفیت محصولات، کاهش زمان عرضه به بازار، و هزینه‌های توسعه می‌شود و همین موضوع دلیل اهمیت و محبوبیت این مدل را روشن می سازد. چالش اصلی در یادگیری مدل خط محصولات نرم‌افزاری، سرعت بالاتر تغییرات و به‌روزرسانی محصولات نسبت به یادگیری ویژگی‌های آن‌ها می‌باشد. در این جاست که یادگیری تطبیقی به کمک می‌آید. در یادگیری تطبیقی، با استفاده از ویژگی‌های حاصل از مدل‌های یادگیری‌شده قبلی می‌توان منجر به افزایش بهینگی و یادگیری محصول فعلی شد. در این پروژه یک روش برای تطبیقی کردن الگوریتم مشهور و مبتنی بر درخت کرنز و وزیرانی [1] ارائه می‌شود. در انتهای این پژوهش می‌توان به سوالات زیر پاسخ داد:

- آیا الگوریتم های مبتنی بر درخت نسبت به دیگر الگوریتم‌های کلاسیک غیردرختی بهینه‌تر عمل می‌کنند؟

[1] Kearns and Vazirani

- آیا امکان پیاده‌سازی یادگیری تطبیقی برای الگوریتم‌های مبتنی بر درخت وجود دارد؟
- در انتخاب ویژگی‌های مورد استفاده برای یادگیری مجدد باید به چه مسائلی توجه شود؟
- میزان کارایی الگوریتم مبتنی بر درخت با اضافه شدن قابلیت یادگیری تطبیقی چه تغییری خواهد کرد؟

۲.۱ مسأله تحقیق

کاربرد و اهمیت مدل توسعه‌ای خط محصولات نرم‌افزار در دنیای امروز غیرقابل انکار است. مسئله مهمی که در مورد مدل‌های رفتاری محصولات خط نرم‌افزار مطرح می‌شود، این است که به دلیل سرعت بالای توسعه، این محصولات به سرعا تغییر می‌کنند و محصولات قبلی از رده خارج می‌شوند. به هیمن دلیل برای تحلیل بیشتر نیاز است تا مدل رفتاری این محصولات از روی پیاده‌سازی‌های قبلی مجددا ساخته بشود. در چنین موقعیتی، ترجیح بر استفاده از یادگیری تظبیقی می‌باشد تا با استفاده از مدل‌های یادگیری شده قبلی، ملزم به اجرای یادگیری از صفر نباشیم. الگوریتم‌های کلاسیک بسیاری در حوزه یادگیری ماشین‌های قطعی تعریف شده‌اند. در این پژوهش، قصد بر آن داریم که افزونگی یادگیری تطبیقی را برای الگوریتم مبتنی بر درخت کرنز و وزیرانی طراحی کنیم و میزان تغییر عملکرد آن را بسنجیم.

۳.۱ تاریخچه‌ای از موضوع تحقیق

مدل‌های رفتاری عنصری اصلی در تحلیل عملکرد محصولات خط نرم‌افزار می‌باشند. الگوریتم‌های کلاسیک بسیاری برای یادگیری مدل‌های رفتاری طراحی شده‌اند اما تمامی آنها به دلیل کم بودن سرعت یادگیری نسبت به سرعت توسعه و به‌روزرسانی محصولات، با چالش اجرای یادگیری از صفر همراه با نیاز به دخالت انسانی روبه‌رو می‌شوند. در این حالت است که یادگیری تطبیقی به کار می‌آید. ایده‌های بسیاری برای یادگیری تحقیقی وجود دارد. در مقاله [۲]، یک روش افزایشی برای یادگیری الگوریتم مبتنی بر درخت کرنز و وزیرانی برای یادگیری ماشین‌های متناهی قطعی ارائه شده است. افزونگی یادگیری تطبیقی تنها برای الگوریتم‌های مبتنی بر درخت کاربرد ندارد، بلکه می‌توان با استفاده از جدول مشاهدات مدل‌های قبلی الگوریتم کلاسیک و غیردرختی L^* را

نیز برای یادگیری ماشین‌های میلی و ماشین‌های متناهی قطعی تطبیقی کرد. [۴، ۱]

۴.۱ تعریف موضوع تحقیق

در این پژوهش، قصد داریم تا الگوریتم مبتنی بر درخت کرنز و وزیرانی را به گونه‌ای تغییر دهیم تا با دریافت یک مجموعه از ورودی جستجوها و خروجی متناظرشان از یادگیری قبلی، مدل فعلی را یادگیری کند. سپس کارایی و عملکرد این الگوریتم تطبیقی را با دیگر الگوریتم‌های تطبیقی و غیرتطبیقی کلاسیک مقایسه می‌کنیم. همچنین به این مسئله می‌پردازیم که جستجوهای به کار رفته در یادگیری مجدد باید چه ویژگی‌هایی داشته باشند. قابل ذکر است که، مدل ورودی تمامی الگوریتم‌های یادگیری در این تحقیق، ماشین‌های حالت متناهی از نوع میلی هستند.

۵.۱ هدف یا هدف‌های کلی و آرمانی تحقیق

همانطور که پیشتر ذکر شد، یادگیری مدل‌های رفتاری محصولات خط نرم‌افزار به دلیل کاربرد گسترده آنها در زمینه‌های مختلف صنعتی، از اهمیتی شایان برخوردار است. همچنین در یادگیری این خانواده از محصولات ترجیح بر استفاده از الگوریتم‌های تطبیقی می‌باشد تا با صرفه‌جویی در تعداد جستجوها، بهینگی یادگیری افزایش یابد و در صورت بروز تغییرات در محصولات جدید با از رده خارج شدن محصولات، نیازی به اجرای یادگیری از پایه نباشد. به این منظور هدف این تحقیق ارائه یک روش برای تطبیقی ساختن الگوریتم مشهور و کلاسیک مبتنی بر درخت کرنز و وزیرانی برای یادگیری ماشین‌های حالت متناهی است. پژوهش‌های این چنینی با تحلیل الگوریتم‌های یادگیری متفاوت، زمینه لازم برای پرورش ایده‌ها در مورد ایجاد تغییر در الگوریتم‌ها و بهبود بخشیدن به آنها را فراهم می‌کنند.

۶.۱ روش انجام تحقیق

اولین قدم در شروع این پژوهش، مطالعه تحقیقات قبلی، تحلیل نتایج آنها و بررسی دقیق الگوریتم‌های کلاسیک می‌باشد. پس از آشنا شدن با نحوه پیاده‌سازی‌های مقالات پیشین، با الهام گیری از مدلسازی و کد پایه مقاله [۱]، اقدام به پیاده‌سازی الگوریتم کلاسیک کرنز و وزیرانی و الگوریتم درختی TTT برای یادگیری ماشین‌های میلی با استفاده از کتابخانه یادگیری ماشین Learnlib کردیم. در قدم بعدی با الهام‌گیری از پژوهش [۴] یک ماشین پیشگو تطبیقی را مدلسازی کردیم تا توسط الگوریتم درختی کرنز و وزیرانی تغییر داده شده، مورد استفاده قرار گیرد. در نهایت با نمایش و بررسی درخت تفاوت مدل‌های یادگیری شده و جستجوهای تولید شده در حین یادگیری آنها، مجموعه‌ای از جستجوهای مدل‌های قبلی را انتخاب کردیم تا در یادگیری مدل جدید و در ماشین پیشگو تطبیقی عضویت، به کار روند. پس از پایان مدلسازی‌ها به اجرای الگوریتم‌های کلاسیک و غیرکلاسیک تطبیقی پرداختیم و با تعریف پارامترهای ارزیابی، به مقایسه کارایی الگوریتم‌ها پرداختیم.

۷.۱ نوآوری، اهمیت و ارزش تحقیق

نوآوری‌های انجام شده در این پژوهش، در دو زمینه قابل بحث می‌باشند.

۱. پیاده‌سازی الگوریتم تطبیقی مبتنی بر درخت کرنز و وزیرانی برای یادگیری ماشین های حالت نامتناهی از جنس ماشین میلی:
در اکثر پژوهش های گذشته یادگیری ماشین‌های قطعی متناهی مورد نظر بوده که در هر حالت با دریافت یک رشته ورودی، یه خروجی از جنس بولین تولید می‌کند که نشان می‌دهد آیا از این حالت با مشاهده این ورودی به حالت پذیرش یا رد کردن می‌رویم. حال آنکه در پژوهش ما الگوریتم‌های یادگیری برای ماشین‌های میلی پیاده‌سازی شده‌اند؛ که در این ماشین‌ها با مشاهده یک رشته از ورودی به جای خروجی صفر و یک، یک رشته از خروجی تولید می‌شود.

۲. تطبیقی ساختن الگوریتم مبتنی بر درخت با کمک استفاده‌ی مجدد از جستجوها:
توجه پژوهش‌های گذشته بیشتر معطوف به الگوریتم‌های کلاسیک و غیر درختی بوده است. [۱، ۴] در مقاله [۲]، نیز که یک الگوریتم تطبیقی برای الگوریتم کرنز و وزیرانی بر روی ماشین‌های قطعی متناهی

تعریف شده، از درخت تفاوت حاصل از یادگیری محصولات قبلی استفاده شده. حال آنکه در این پژوهش الگوریتم تطبیقی برای الگوریتم کلاسیک و مبتنی بر درخت کرنز و وزیرانی به منظور یادگیری ماشین‌های میلی ارائه می‌شود. در این الگوریتم از جستجوهای مدل قبلی در یادگیری مجدد استفاده می‌شود.

۸.۱ تعریف واژه‌ها

در این بخش به توضیح اصطلاحات و مفاهیم مهم استفاده شده در این تحقیق می‌پردازیم.

- خط محصولات نرم‌افزار [۲]

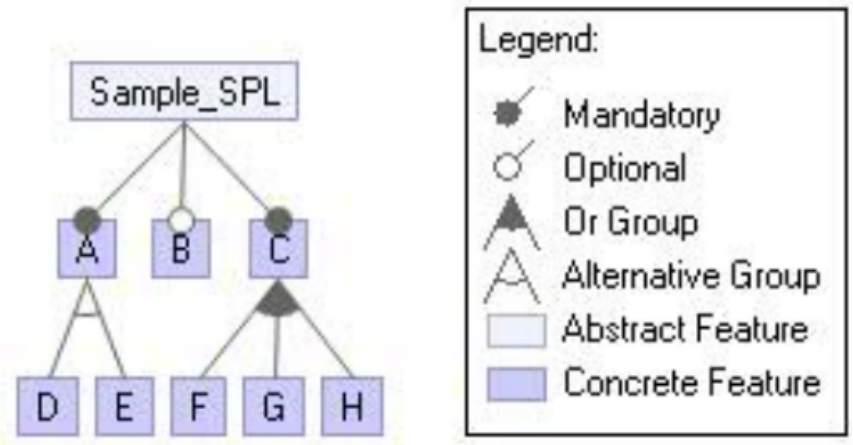
خط محصولات نرم‌افزار یک رویکرد توسعه می‌باشد که بر تولید خانواده‌هایی از محصولات نرم‌افزاری که مجموعه‌ای از ویژگی‌ها و مشخصات مشترک دارند تمرکز می‌کند. در این نوع از محصولات یک هسته مرکزی از کاربردها طراحی و پیاده‌سازی می‌شود و سپس شخصی‌سازی می‌شود تا نیازهای یک محصول مشخص را برآورده کند. هدف از طراحی خط محصولات نرم‌افزار بهبود کارایی و کیفیت توسعه نرم‌افزار به وسیله استفاده مجدد از اجزا و ویژگی‌های مشترک و کاهش افزونگی ها [۳] می‌باشد.

برای پیاده‌سازی یک خط محصول نرم‌افزار یک معماری با دامنه مشخص تعریف و طراحی می‌شود که اشتراکات و متغیرهای محصولات نرم‌افزاری را تعریف می‌کند. هر مدل خط محصول نرم‌افزار معمولا با یک مجموعه از ویژگی‌ها و یک مدل ویژگی [۴] تعریف می‌شود. یک مدل ویژگی ساختاری سلسله مراتبی از ویژگی‌های محصول است.هر کدام از محصولات این خط محصول نرم‌افزار از زیرمجموعه‌ای از این ویژگی‌ها تشکیل می‌شوند. تنظیمات قابل قبول برای هر محصول توسط مدل ویژگی آن مشخص می‌شود. در این نمایش ویژگی‌ها به دو دسته اختیاری و اجباری تقسیم می‌شوند. ویژگی‌های اجباری به صورت خودکار در تمام محصولات خط نرم‌افزار حضور دارند. از هر مجموعه ویژگی‌های انتخابی فقط یکی می‌تواند در محصول وجود داشته باشد.

تصویر ۱.۱ نمایی از یک مدل ویژگی یک خط محصول نرم‌افزاری نمونه می‌باشد. در این تصویر ویژگی A و C اجباری و ویژگی B اختیاری می‌باشند. ویژگی‌های D و E انتخابی هستند. ویژگی‌های F، G و

[2] Software Product Lines (SPL)
[3] Redundancy
[4] Feature Model

H یک گروه فصلی از ویژگی‌ها تشکیل می‌دهند. این خط محصول نرم‌افزار حاوی ۲۸ ترکیب معتبر از ویژگی‌های محصول می‌باشد.



شکل ۱.۱: یک مدل ویژگی نمونه

- ماشین‌های حالات متناهی [5]

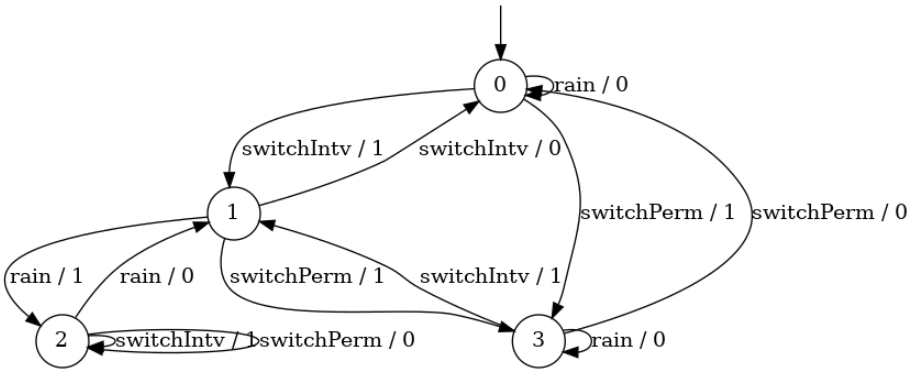
ماشین حالات متناهی یک مدل رفتاری پرکاربرد می‌باشد که نحوه انتقال حالت‌های مختلف ماشین را بسته به ورودی‌های دریافتی نمایش می‌دهد. این مدل تعدادی متناهی از حالات و ورودی‌ها را دارد. در هر زمان ماشین حالت نامتناهی فقط در یکی از حالت‌ها می‌باشد. پس از مشاهده هر ورودی، ماشیت حالت بسته به حالت کنونی‌اش و ورودی دریافتی به یک حالت جدید می‌رود. هر انتقال بین حالات یک خروجی مشخص دارد که ماشین تولید می‌کند.

ماشین حالت متناهی دو نوع قطعی و غیر قطعی دارد. در مدل قطعی، به ازای هر حالت و هر ورودی فقط یک انتقال یکتا داریم؛ در حالی که در مدل غیرقطعی، هر حالت به ازای هر ورودی ممکن است انتقال‌های متفاوتی داشته باشد. در این پژوهش هر جا به ماشین حالت متناهی اشاره شده، منظور ماشین قطعی است. هر ماشین حالت متناهی توسط یک مجموعه چندتایی به صورت $M = \langle S, s_\circ, I, O, \delta, \gamma \rangle$ تعریف می‌شود. در این تعریف S مجموعه‌ای از حالات و $s_\circ$ حالت اولیه $s_\circ \in S$ می‌باشد. I الفبای ورودی و O نشاندهنده‌ی مجموعه‌ای از خروجی‌هاست. تابع انتقال‌های بین حالات δ حالت بعدی را مشخص

[5] Finite State Machines (FSMs)

می‌کند $s_۲ \in S$. با فرض اینکه اتوماتا در حالت $s_۱ \in S$ و الفبای ورودی $a \in I$، تابع انتقال به صورت $((\delta s_۱, a) = s_۲)$ نمایش داده می‌شود. تابع خروجی با γ نمایش داده می‌شود که یک نگاشت از جفت ورودی و حالت به یک حالت نهایی است. ماشین‌های حالت یادگیری‌شده توسط مدل‌های قبلی در این پژوهش از نوع قطعی می‌باشند که در به ازای هر حالت و ورودی حداکثر یک انتقال و یک خروجی خواهند داشت. [۴]

تصویر ۲.۱ نمایه‌ای از یک ماشین حالت متناهی از یک سیستم برف پاککن ماشین است که در این پژوهش نیز به کارگرفته شده است.



شکل ۲.۱: یک ماشین حالت متناهی نمونه

۹.۱ خلاصه فصل‌ها

در این فصل به مقدمه‌ای از این پژوهش و تعریف مسئله مورد تحقیق پرداخته شد. همچنین توضیحاتی در رابطه با اصطلاحات و واژه‌های مورد استفاده در این پژوهش ارائه شد. در پایان، نحوه انجام تحقیق به طو کلی و نوآوری‌های تحقیق ذکر شده‌اند. در فصل دوم، تاریخچه‌ای از پژوهش‌های انجام شده در این زمینه که الهام بخش پژوهش ما بودند ارائه می‌شود. فصل سوم به روش انجام تحقیق، مدلسازی‌ها، پیاده‌سازی‌ها و ابزارهای مورد استفاده در یادگیری به تفصیل می‌پردازد. در فصل چهارم نتایج حاصل از بررسی‌ها مورد بحث قرار خواهند گرفت. در آخرین فصل از این پژوهش تحلیل‌ها و نتایج به دست آمده جمع‌بندی می‌شود و برخی مسیرهای آینده برای سایر محققان ذکر خواهد شد.

فصل ۲

مروری بر مطالعات انجام شده

۱.۲ مقدمه

مدل‌های رفتاری عنصری اصلی در تحلیل عملکرد محصولات خط نرم‌افزار از جمله آزمون و وارسی مدل‌های رفتاری [۱] هستند. مدل‌های کلاسیک مشهور یادگیری فعال مانند L*، Kearns and Vazirani و TTT برای تعدادی از انواع اتوماتاها از جمله اتوماتاهای قطعی [۲] ، ورودی-خروجی [۳]، وزن‌دار [۴] و رجیستری [۵] تعریف شده‌اند و امکان بررسی تعداد قابل‌توجهی از سیستم‌ها را با ارائه مدل‌های قابل اعتماد برای استفاده در وارسی مدل‌ها فراهم آورده‌اند. اما تمامی این الگوریتم‌ها مشکلی مشابه دارند: سیستم‌ها معمولا سریع‌تر از این که ما قادر به یادگیری تمام ویژگی‌های مدل‌های رفتاریشان باشیم به روزرسانی می‌شوند و تمامی الگوریتم‌های مذکور برای هر بار یادگیری مدل باید از پایه اجرا شوند. در نتیجه به روز نگه داشتن مدل رفتاری یک سیستم نیازمند دخالت دستی زیاد از سوی یک متخصص این حوزه می‌باشد که بسیار چالش‌برانگیز است. در این حالات امکان تولید محصولاتی که دقیقا عملکرد صحیح و قابل‌انتظار را ندارد بالا می‌رود. [۲]

در این حالت است که یادگیری تطبیقی به کار می‌آید. به دلیل اشتراکات زیاد بین محصولات خط نرم‌افزار، زمینه استفاده از روش‌های یادگیری تطبیقی به وجود می‌آید. به این صورت که در حالاتی که مدل رفتاری یک

[1] Behavorial Model Checking
[2] Deterministic
[3] Input-Output
[4] Weighted
[5] Register

خانواده از محصولات وجود ندارد یا به‌روز نیست می‌توان از روش یادگیری تطبیقی و دیگر محصولات مشابه این خط محصول استفاده کرد. در این بخش به بررسی پیشینه این زمینه از تحقیق و مقالات چاپ شده مرتبط که الهام‌بخش این پروژه بودند می‌پردازیم.

۲.۲ مروری بر ادبیات موضوع

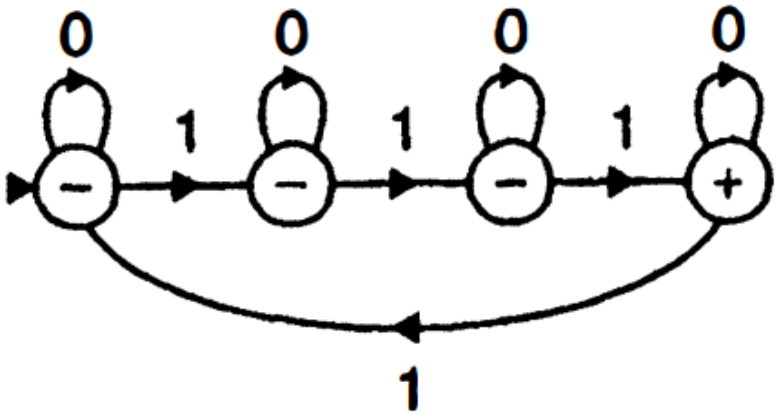
تحقیقاتی که الهام‌بخش این پژوهش بودند در سه دسته‌ی، یادگیری کلاسیک [۳] ، یادگیری تطبیقی [۴، ۵، ۱] و یادگیری افزایشی [۲] جای می‌گیرند.

- یادگیری کلاسیک

در این تحقیق برای اولین بار یک روش یادگیری دقیق ماشین‌های قطعی متناهی بر اساس جستجوهای عضویت و معادلسازی معرفی می‌شود که در نهایت منجر به الگوریتم پایه کرنز و وزیرانی می‌شود. ایده‌ی اصلی این الگوریتم استفاده از یک ساختار داده به نام درخت تفاوت به منظور یادگیری اتوماتاهای قطعی متناهی از روی مجموعه‌ای از مثال‌های مثبت و منفی است. درخت تفاوت تمامی فضای فرضی که الگوریتم تصمیم به یادگیری آن دارد را ذخیره می‌کند. این درخت به صورت بازگشتی مجموعه مثال‌های مثبت و منفی را به زیرمجموعه‌های کوچکتری بر اساس مقادیر مشاهده شده از ورودی دسته‌بندی می‌کند. در هر سطح از درخت، الگوریتم یک گره جدید تولید می‌کند که مربوط به دسته‌بندی مثال‌ها در این مرحله از یادگیری می‌باشد. این گره برچسبی دارد که نشان‌دهنده‌ی مجموعه‌ای از ورودی‌هاست که تا به حال برای این دسته از مثال‌ها دیده شده است. درخت تفاوت به نوعی تولید می‌شود که تضمین می‌کند که هر گره در این درخت مربوط به اتوماتا قطعی متناهی است که با مثال‌های موجود در این دسته سازگار [۶] است. این الگوریتم از ریشه درخت که نشاندهنده‌ی عمومی ترین اتوماتا قطعی متناهی ممکن که تمام رشته‌های ورودی را می‌پذیرد (اتوماتای پذیرنده رشته اپسیلون) است، شروع می‌کند. در روند ساخت درخت تفاوت، الگوریتم فرض خود را از اتوماتای قطعی متناهی هدف یا استفاده از جستجوهای عضویت و معادل‌سازی بازتعریف می‌کند. به این صورت که در هر مرحله هر کدام از اتوماتاهای قطعی نامتناهی که با دسته‌بندی متناظرشان ناسازگار هستند را حذف می‌کند (مثال‌های نقض). این مراحل آنقدر تکرار می‌شوند تا در

[6] Consistent

یک برگ درخت به اتوماتای قطعی متناهی هدف برسیم و الگوریتم متوقف می‌شود. تصویر ۱.۲ مثالی از اجرای این الگوریتم است. در این مثال اتوماتای قطعی نامتناهی یک رشته را می‌پذیرد اگر و تنها اگر تعداد یک‌ها در این رشته به پیمانه چهار برابر سه باشد.



شکل ۱.۲: اتوماتای قطعی که در رشته $3 \mod 4$ تعداد یک‌ها را می‌شمارد. [۲]

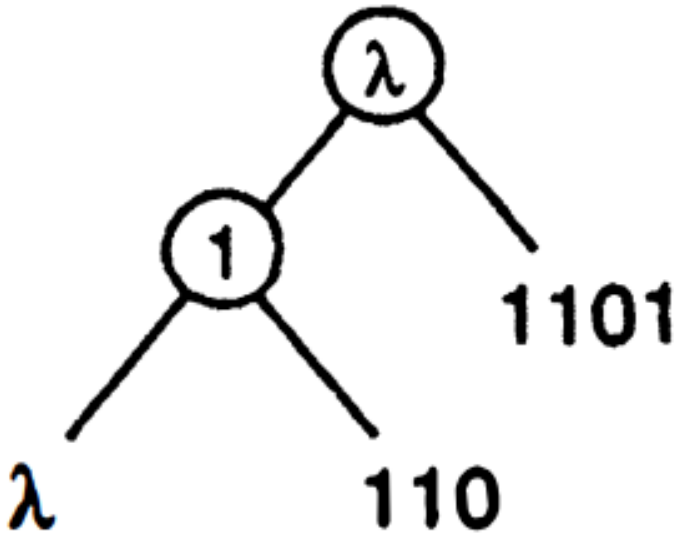
در تصویر ۲.۲ درخت تفاوت این اتوماتا نمایش داده شده است. در این درخت فرزندان راست رشته‌های قابل قبول هستند.

مزیت اصلی استفاده از درخت تفاوت، نمایش و ذخیره فشرده فرض مدل است. اندازه این درخت با لوگاریتم اندازه فرض متناسب است و همین ویژگی باعث بهینه و کارا بودن این الگوریتم نسبت به دیگر الگوریتم‌های کلاسیک می‌شود.

در پژوهش پیش‌رو قصد بر این داشتیم تا با تطبیقی کردن این مدل از یادگیری کارایی آن را برای یادگیری محصولات خط نرم‌افزار افزایش دهیم.

- یادگیری تطبیقی

یادگیری تطبیقی یک افزونه بر روش‌های یادگیری سنتی است که با استفاده از مدل‌های قبلی موجود کارایی و بهینگی الگوریتم‌های سنتی را بهبود می‌بخشد. یادگیری تطبیقی زمانی بهترین نتیجه را می‌دهد که محصولات خط نرم‌افزار به‌روزرسانی‌های محدودی نسبت به محصولات قبلی در داشته باشند. در مقاله



شکل ۲.۲: درخت تفاوت متناظر این اتوماتا [۲]

یادگیری تطبیقی مدل‌های رفتاری برای محصولات خط نرم‌افزار [۴]، الگوریتم تطبیقی *PL معرفی شده است که بر پایه‌ی الگوریتم شناخته شده و کلاسیک *L می‌باشد. در الگوریتم *L از یک ساختار داده به نام جدول مشاهدات استفاده می‌شود. این جدول از دو بخش تشکیل شده‌است؛ بخش بالایی جدول که حاوی مجموعه‌ای از پیشوندهای رشته ورودی می‌باشد و بخش پایینی جدول که از پسوندهای رشته‌ی ورودی تشکیل شده است. هر موجودیت در این جدول نشانگر یک جفت از ورودی و خروجی‌هاست که برای ماشین میلی هدف مشاهده شده‌اند. جدول مشاهدات به صورت مرحله به مرحله همانطور که الگوریتم، جستجوهای معادلسازی را انجام می‌دهد ساخته می‌شود. الگوریتم *L از این جدول استفاده می‌کند تا فرض خود از ماشین میلی هدف را بهبود ببخشد و نهایتا به مدل نهایی سازگار برسید. در الگوریتم *PL تعریف‌شده در مقاله [۴] پس از یادگیری محصول ابتداییه پسوند و پیشوندهای جدول مشاهدات آن برای یادگیری محصول بعدی به کار می‌رود تا تعداد جستجوها یادگیری کاهش یابد.

همچنین در پژوهش [۱] یک الگوریتم تطبیقی به نام الگوریتم *L پویا برای بهبود کارایی الگوریتم کلاسیک *L معرفی شده است. در این روش به صورت آنی [۷] جدول مشاهدات حاصل از یادگیری مدل قبلی بررسی می‌شود تا تمامی پسوند و پیشوندهای اضافه موجود در جدول حذف شوند. به این

[7] On the Fly

صورت با جدول بهینه شده تعداد جستجوهای عضویت کاهش می‌یابد.

با الهام‌گیری از ایده این مقاله‌ها، الگوریتم تطبیقی کرنز و وزیرانی که حاوی درخت تفاوت است در پژوهش پیش‌رو پیاده‌سازی شده است.

- یادگیری افزایشی

در مقاله، [۲] بر خلاف روش تطبیقی که از جستجوهای مدل محصولات پیشین استفاده می‌کند، یک روش افزایشی برای الگوریتم مبتنی بر درخت کرنز و وزیرانی معرفی شده است. در این الگوریتم، مدل محصول ابتدایی به صورت کلاسیک یادگیری شده و سپس درخت تفاوت این مدل برای یادگیری محصولات بعدی استفاده می‌شود. به این صورت که برای محصول بعدی مدل به جای شروع از گره خالی از درخت مدل پیشین استفاده می‌کند. قبل از شروع یادگیری لازم است که این درخت پیمایش شود تا برگ‌هایی که برای یادگیری این محصول قابل استفاده نیستند، هرس شوند. این مرحله آنقدر ادامه پیدا می‌کند تا به زیرمجموعه‌ای از درخت اولیه برسیم که با مدل محصول کنونی سازگار باشد. پس از ساخت کامل درخت، یادگیری آغاز می‌شود. بدین صورت با شروع از یک حد واسط درختی به جای گره خالی کارایی الگوریتم بهبود می‌یابد و الگوریتم برای یادگیری محصولات خط نرم‌افزار که تفاوت کمی باهم دارند مناسب خواهد شد.

خللی که در این پژوهش وجود دارد این است که این الگوریتم تنها برای ماشین‌های قطعی متناهی پیاده سازی شده است و راه‌حلی برای ماشین‌های میلی ارائه نمی‌کند.

۳.۲ نتیجه‌گیری

با توجه به توضیحات ارائه‌شده در مورد تحقیقات پیشین و بررسی خلل‌های آنها تصمیم بر این گرفته شد تا پژوهشی در حوزه یادگیری تطبیقی مبتنی بر درخت بر روی ماشین‌های میلی [۸] و سیستم های طراحی شده بر پایه ماشین‌های حالات متناهی [۹] انجام دهیم و به بررسی و مقایسه این الگوریتم با مدل‌های پیشین بپردازیم.

[8] Mealy Machines
[9] Finite State Machines (FSM)

فصل ۳

راه‌حل پیشنهادی

۱.۳ مقدمه

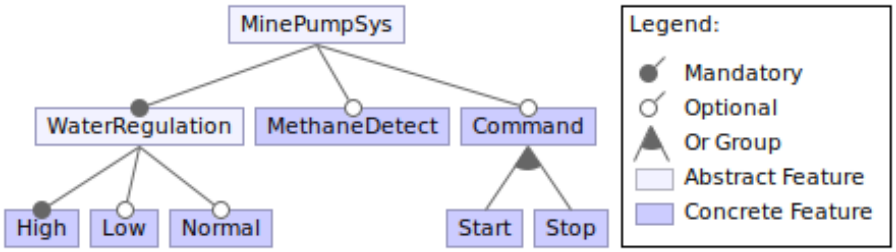
به منظور رفع خلل‌های موجود در زمینه یادگیری تطبیقی مبتنی بر درخت که در فصول قبلی به آنها اشاره شد، تصمیم به انجام این پژوهش شد. در این فصل به توضیح روش استفاده‌شده در این تحقیق به طور کامل می‌پردازیم.

۲.۳ تشریح کامل روش تحقیق

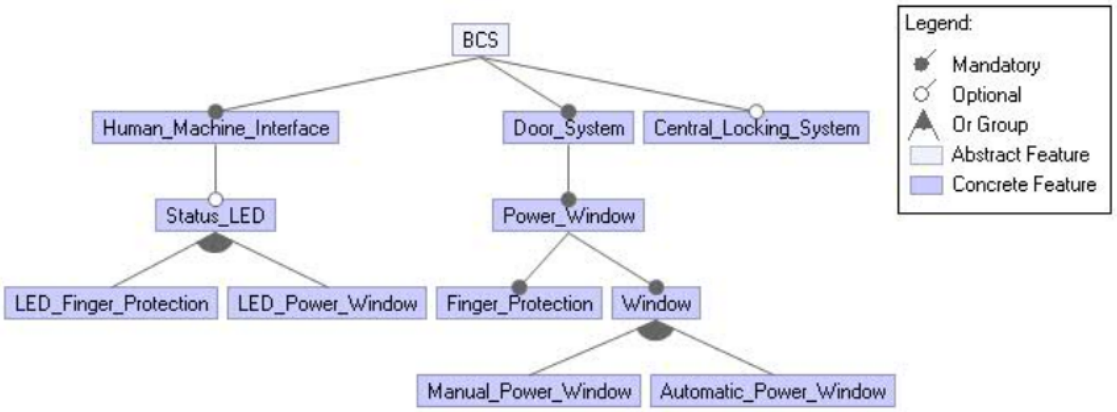
۱.۲.۳ ماشین‌های حالات قطعی

برای این پژوهش از دو خط محصول نرم‌افزار معروف و متن‌باز Minepump (MP) و Body Comfort System (BCS) استفاده کرده‌ایم. تمامی محصولات این خطوط نرم‌افزار به صورت ماشین‌های حالات قطعی ذخیره شده‌اند و به عنوان ورودی به برنامه داده شده‌اند. هر کدام از این خطوط محصولات نرم‌افزاری حاوی یک مدل ویژگی است. بر اساس تنظیمات موجود در این مدل ویژگی، ترکیبات مجاز از ویژگی‌هامشخص می‌شوند و به این ترتیب تعداد مشخصی از ماشین‌های حالات قطعی برای هر خط محصول به وجود میایند.

مدل‌های ویژگی این خط محصول نرم‌افزاری در تصاویر ۱.۳ و ۲.۳ نمایش داده شده‌اند.



شکل ۱.۳: مدل ویژگی خط محصولات نرم‌افزاری Minepump



شکل ۲.۳: مدل ویژگی خط محصولات نرم‌افزاری Body Comfort System

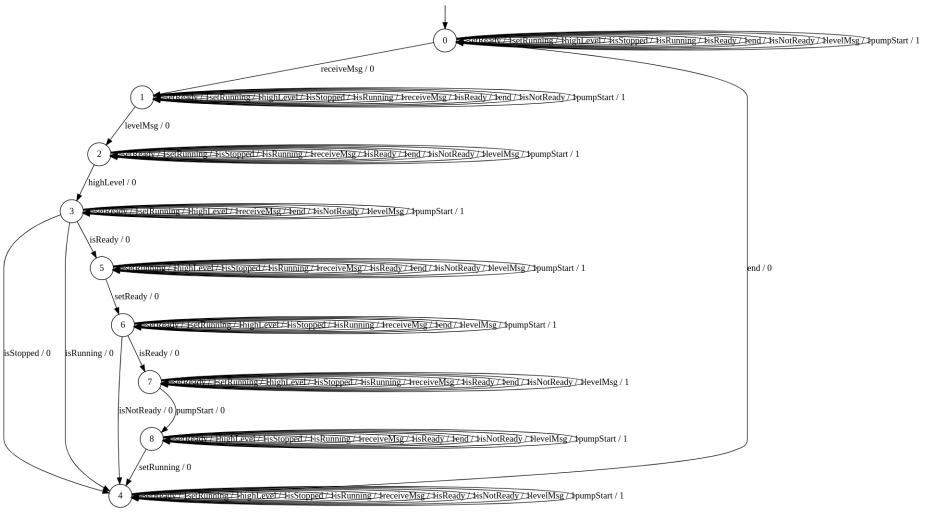
در این مجموعه، هر محصول از طریق دو فایل که فایل اول حاوی ویژگی‌هاو فایل دوم حاوی ماشین حالت متناهی محصول می‌باشد، معرفی می‌شود.

به عنوان مثال یکی از محصولات MP، دارای ویژگی‌های $\{MinePumpSys, WaterRegulation, High\}$ می‌باشد که در فایل تنظیمات نوشته شده‌اند. ماشین حالت این محصول در تصویر ۳.۳ نشان داده شده است.

۲.۲.۳ جستجوها

دو جز اصلی در مسائل یادگیری اتوماتا ماشین‌های پیشگو [1] عضویت و معادلسازی هستند که برای یادگیری ماشین هدف به کار می‌روند.

[1] Oracle



شکل ۳.۳: ماشین حالت متناهی محصول نمونه MP

- ماشین پیشگو عضویت مسئول پاسخگویی به جستجوها در مورد رفتار ماشین هدف است. در پژوهش ما که نوع ماشین مورد مطالعه از نوع ماشین میلی می‌باشد، ماشین پیشگو عضویت یک رشته ورودی را دریافت می‌کند و جواب ماشین فرضی هدف را به این رشته به عنوان خروجی تولید می‌کند. ماشین پیشگو عضویت می‌تواند به دو نوع جعبه-سیاه یا جعبه-سفید پیاده‌سازی شود. در مدل جعبه-سیاه پیشگو رفتار ماشین هدف را شبیه‌سازی می‌کند در حالی که، در مدل جعبه-سفید پیشگو به محتویات حالت داخلی سیستم هدف دسترسی کامل دارد. در این پژوهش از مدل ماشین پیشگو عضویت جعبه-سیاه استفاده شده است.

- ماشین پیشگو معادلسازی مسئول بررسی این می‌باشد که آیا ماشین فرضی تولید شده یک ماشین زبانی معادل برای ماشین هدف می‌باشد یا خیر. این پیشگو یک ماشین فرضی [۲] به عنوان ورودی دریافت می‌کند و اگر این فرض با ماشین هدف مطابقت نداشته باشد، یک مثال نقض برمی‌گرداند و اگر به ماشین هدف رسیده باشیم، یادگیری پایان می‌یابد. پیشگو معادلسازی انواع متفاوتی دارد که در زیر به دو نوع مشهور آن اشاره می‌کنیم.

Random Walk EQOracle -

در این نوع پیشگو به صورت تصادفی فضای نمونه را بررسی می‌کند و یک رشته ی ورودی تصادفی

[2] hypothesis automaton

بر اساس الفبای مدل تولید می‌کند و پاسخ ماشین هدف و فرض را به این ورودی با هم مقایسه می‌کند.

- W-Method EQOracle

این پیشگو از روش W استفاده می‌کند و به جای تولید تصادفی رشته ی ورودی با اطلاعاتی که از هر دو ماشین فرض و هدف دارد، رشته‌ای را تولید می‌کند که احتمال بالایی در تشخیص تفاوت دو ماشین دارد. به این صورت در زمانی بسیار کمتر و به صورت بهینه‌تر مثال نقض را پیدا می‌کند.

در این پژوهش از پیشگو بهینه‌تر W-Method EQOracle استفاده شده است.

۳.۲.۳ الگوریتم‌های پیشین

به منظور مقایسه الگوریتم تطبیقی پیشنهادی در این پژوهش، ابتدا الگوریتم‌های پیشین مورد نظر خود را پیاده‌سازی کردیم. این الگوریتم‌ها عبارتند از:

- الگوریتم Kearns and Vazirani[۲]

نحوه یادگیری این الگوریتم در بخش ۲.۲ با جزییات ارائه شده است. این الگوریتم برای یادگیری از یک درخت تفاوت استفاده می‌کند که آن را بر اساس خروجی‌های حاصل از ماشین پیشگو عضویت و معادلسازی تولید می‌کند و تا رسیدن به ماشین هدف، فرض خود را با استفاده از اطلاعات موجود در درخت بازتعریف می‌کند. برای مدلسازی این الگوریتم از پیاده سازی آن در کتابخانه Learnlib استفاده کردیم.

- الگوریتم TTT

مشابه الگوریتم کرنز و وزیرانی، این الگوریتم نیز یک الگوریتم مبتنی بر درخت می‌باشد که برای یادگیری اتوماتا استفاده می‌شود. این الگوریتم با اضافه کردن پیوسته حالت‌ها و انتقال‌ها به ماشین فرضی با استفاده از مجموعه‌ای از هیورستیک‌ها که انتقال‌های بین حالت‌ها را اولویت‌دهی می‌کنند، یادگیری را انجام می‌دهد. ایده اصلی این الگوریتم در استفاده از یک مجموعه آزمایش تصادفی از رشته‌های ورودی می‌باشد که دقت فرض را ارزیابی می‌کنند و تا رسیدن به ماشین هدف و دسته‌بندی صحیح تمام ورودی‌ها،

فرض را بازتعریف می‌کنند. برای مدلسازی این الگوریتم از پیاده سازی آن در کتابخانه Learnlib استفاده کردیم.

- الگوریتم *L

برای پیاده سازی این الگوریتم از طراحی آن در کتابخانه Learnlib استفاده کردیم. این الگوریتم کلاسیک با استفاده از جدوا مشاهدات تولید شده بر اساس خروجی‌های ماشین پیشگو عضویت و معادلسازی به یادگیری ماشین هدف می‌پردازد.

- الگوریتم *PL

برای پیاده‌سازی این الگوریتم از طراحی آن در مقاله [۴] استفاده کردیم که با دریافت پسوند و پیشوندهای جدول مشاهدات حاصل از یادگیری قبلی جدول مشاهدات محصول را به روز رسانی می‌کند تا در تعداد جستجوها صرفه‌جویی شود.

۴.۲.۳ نمایش درخت تفاوت

برای اطمینان حاصل کردن از استفاده درست الگوریتم کلاسیک و همچنین برای دسترسی به پیشوندو پسوندهای استفاده شده در یادگیری نیاز به مشاهده و تحلیل درخت تفاوت الگوریتم مبتنی بر درخت می‌باشد. درخت تفاوت استفاده شده در الگوریتم کرنز و وزیرانی از نوع Multi D-Tree می‌باشد که ابزار مستقیمی برای نمایش آن وجود نداشت. برای حل این مسئله ابتدا الگوریتمی طراحی کردیم که با شروع از ریشه‌ی درخت، فرزندان هر نود را به دست آورده و به فرمت dot در یک فایل ذخیره کردیم. سپس با استفاده از ابزار Graph Viz این درخت نمایش داده شد. در تصویر ۴.۳ و ۵.۳ نمایی از درخت تفاوت تولید شده و نمایش متنی از جنس dot آن مشاهده می‌شود. هر گره در این درخت نمایشگر کلمه ورودی دیده‌شده و هر یال نمایشگر رشته خروجی تولیدشده توسط ماشین میلی فرضی می‌باشد.

۵.۲.۳ الگوریتم تطبیقی

ایده‌ی اصلی پیاده‌سازی الگوریتم تطبیقی کرنز و وزیرانی در این پژوهش، تطبیقی کردن ماشین پیشگو عضویت می‌باشد. برای مدلسازی این ماشین یک کلاس پیشگو جدید به صورت لفافه‌بندی [۳] طراحی کردیم که ماشین پیشگو اصلی را پیاده‌سازی [۴] می‌کند و ماشین پیشگو عضویت را به عنوان یک فیلد پوشش می‌دهد. حال برای کاهش تعداد جستجوها از یک نگاشت [۵] کلمه به کلمه استفاده کردیم که ورودی هر جستجو را به خروجی متناظرش می‌نگارد. این نگاشت پیش از شروع یادگیری توسط جستجوهای مدل قبلی مقداردهی می‌شود. حال با بازنویسی متد AnswerQuery از کلاس Mealy Membership Oracle از نگاشت طراحی شده استفاده می‌کنیم به این صورت که چنانچه ورودی جستجو در نگاشت ما موجود بود، رشته متناظر خروجی آن به عنوان پاسخ جستجو بازگردانده می‌شود و اگر این جستجو در نگاشت طراحی شده موجود نباشد، متد پاسخ جستجو از کلاس ماشین پیشگو عضویت فراخوانی می‌شود تا پاسخ جستجو بازگردانده شود.

۶.۲.۳ ویژگی‌های انتخابی برای یادگیری مجدد

برای انتخاب ویژگی‌های مورد استفاده در یادگیری مجدد، باید به چند مسئله توجه کرد.

- انتخاب جستجوها

ورودی جستجوهای حاصل از یادگیری محصول قبلی باید در الفبای محصول جدید موجود باشند. همچنین انتقال‌های بین حالتی مناسب‌تر از انتقال‌های طوقه‌ای هستند. در نتیجه برای انتخاب جستجوها به ازای هر کلمه بررسی می‌کنیم که آیا انتقال حاوی این کلمه در الفبای محصول ما موجود است یا خیر و اگر موجود بود، کلمه برای یادگیری مجدد انتخاب می‌شود.

- انتخاب محصولات

یادگیری تطبیقی زمانی بهترین نتیجه را می‌دهد که محصولات خط نرم‌افزار تغییرات و به روزرسانی محدودی نسبت به هم داشته باشند. دو محصول از یک خط محصولات نرم‌افزاری به نام‌های الف و

[3] Wrapper
[4] Implementation
[5] Mapping

ب در نظر بگیرید. در این پژوهش، محصول الف برای یادگیری محصول ب استفاده می‌شود اگر محصول الف زیرمجموعه ای از محصول ب باشد. در این حالت است که بهبود یادگیری قابل‌انتظارتر خواهد شد.

۳.۳ ابزارهای استفاده شده در تحقیق

۱.۳.۳ کتاب‌خانه یادگیری Learnlib

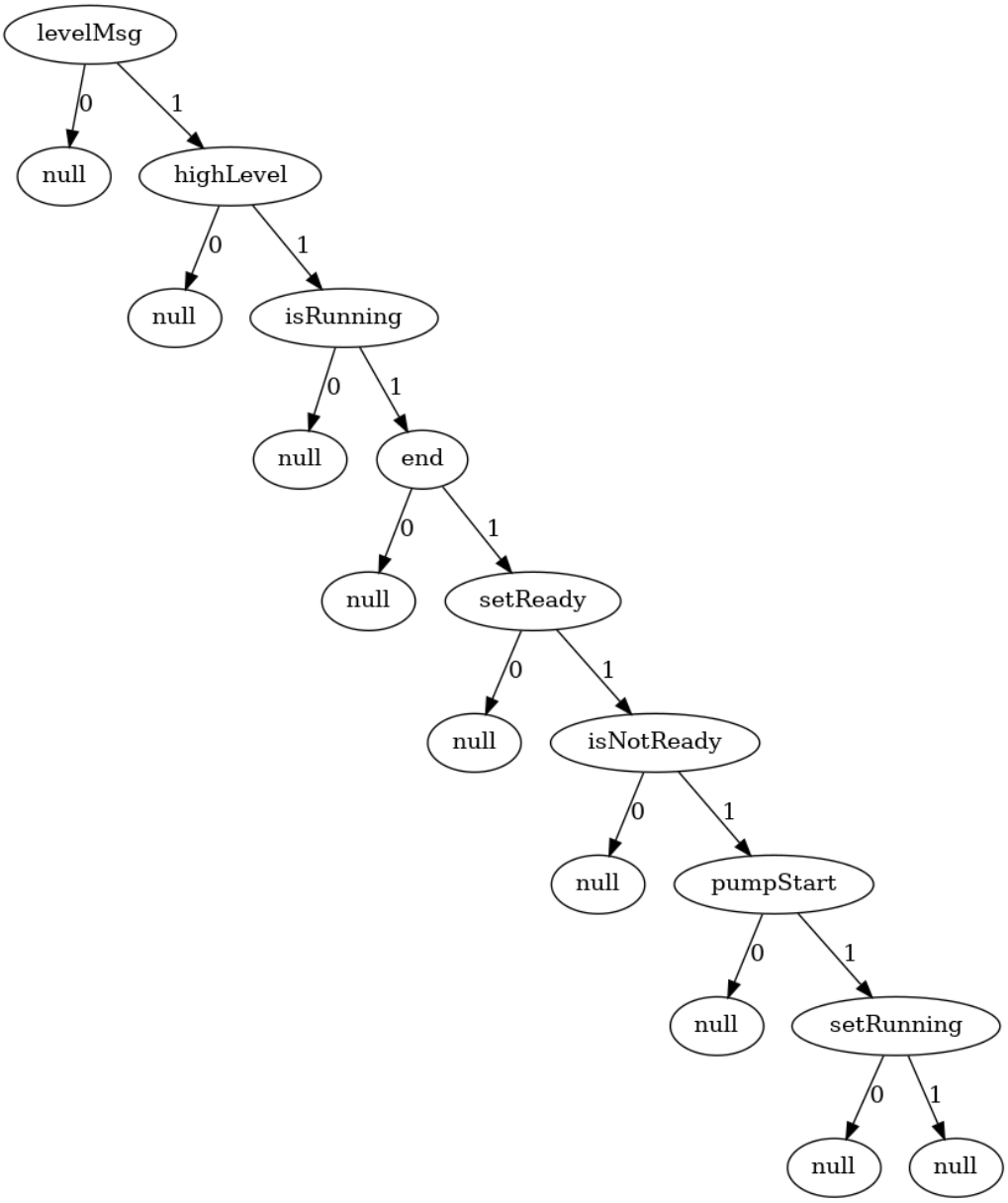
برای مدلسازی این پژوهش از پیاده‌سازی‌های موجود در کتاب‌خانه Learnlib استفاده کردیم. Learnlib یک کتاب‌خانه نرم‌افزاری متن‌باز برای یادگیری اتوماتاها می‌باشد. این کتابخانه مجموعه‌ای ابزارها و الگوریتم‌ها برای یادگیری اتوماتا از جمله الگوریتم‌های یادگیری‌های فعال و منفعل، ماشین‌های پیشگو معادلسازی و عضویت و ابزارهای بررسی مدل آماری را ارائه می‌کند.

۲.۳.۳ نمایش گراف‌ها

برای نمایش ساختارهای گرافی در این پژوهش از بسته نرم‌افزاری متن‌باز Graphviz استفاده کردیم. هسته مرکزی این نرم‌افزار از مجموعه‌ای از ابزارهای خط فرمان برای ساخت و نمایش گراف‌ها تشکیل شده است. تمامی این ابزارها از یک زبان ساده به نام DOT برای مشخص کردن ساختار گراف، گره‌ها و یال‌ها استفاده می‌کنند. همچنین این بسته حاوی مجموعه‌ای از ابزارات برای ایجاد تغییرات در ساختارهای گرافی و آنالیز آنها می‌باشد.

۳.۳.۳ کدبیس مقاله الگوریتم *DL [۱]

تمامی پیاده‌سازی‌های انجام شده در این پژوهش در ادامه کد پایه مقاله [۱] ، انجام شده است. در پژوهش یادشده الگوریتم کلاسیک *L و نسخه‌هایی متفاوتی از الگوریتم تطبیقی *DL معرفی شده‌اند که همگی از کتابخانه یادگیری Learnlib استفاده می‌کنند.



شکل ۴.۳: نمایش درختی درخت تفاوت

```
digraph G {
0[label="levelMsg"];
0 -> 1 [label="0"];
0 -> 2 [label="1"];
1[label="null"];
2[label="highLevel"];
2 -> 3 [label="0"];
2 -> 4 [label="1"];
3[label="null"];
4[label="isRunning"];
4 -> 5 [label="0"];
4 -> 6 [label="1"];
5[label="null"];
6[label="end"];
6 -> 7 [label="0"];
6 -> 8 [label="1"];
7[label="null"];
8[label="setReady"];
8 -> 9 [label="0"];
8 -> 10 [label="1"];
9[label="null"];
10[label="isNotReady"];
10 -> 11 [label="0"];
10 -> 12 [label="1"];
11[label="null"];
12[label="pumpStart"];
12 -> 13 [label="0"];
12 -> 14 [label="1"];
13[label="null"];
14[label="setRunning"];
14 -> 15 [label="0"];
14 -> 16 [label="1"];
15[label="null"];
16[label="null"];
}
```

شکل ۵.۳: نمایش متنی درخت تفاوت

فصل ۴

نتایج

۱.۴ مقدمه

تا اینجا در فصل اول، مقدمه‌ای از یادگیری مدل‌های رفتاری محصولات خط نرم‌افزار و اهمیت بررسی آن‌ها مورد بحث قرار گرفت. در فصل دوم، ادبیات تحقیق و تاریخچه‌ای از کارهای پیشین ارائه شد. در فصل سوم، خط محصولات نرم‌افزاری استفاده شده در این پژوهش معرفی شدند. سپس به تفصیل الگوریتم‌های کلاسیک و نحوه مدلسازی الگوریتم تطبیقی کرنز و وزیرانی توضیح داده شد. در این فصل، به بررسی نتایج تحفیف و پاسخ دادن به سوالاتی که در ابتدا مطرح کرده بودیم، می‌پردازیم. ابتدا شاخص‌های ارزیابی یادگیری را معرفی می‌کنیم و سپس نتایج یادگیری الگوریتم‌های کلاسیک و تطبیقی بر روی محصولات خط نرم‌افزاری BCS و MP تحلیل خواهند شد.

۲.۴ شاخصه‌های اعتبار سنجی

هر یادگیری با شاخصه‌های زیر تعریف می‌شود، که در این بخش به توضیح آنها می‌پردازیم.

- تعداد بازنشانی‌های جستجوهای عضویت (MQ Resets)

در حین یادگیری، پس از تعداد مشخصی از جستجوها، تعداد جستجوها به صفر بازنشانی می‌شود. این

تکنیک باعث می‌شود تا الگوریتم در کمینه محلی گیر نکند و بتواند مناطق متفاوتی از فضای فرض را بررسی کند.

- تعداد بازنشانی‌های جستجوهای معادلسازی (EQ Resets)

این مشخصه نیز مشابه بازنشانی‌های جستجوهای عضویت عمل می‌کند تا ماشین تحت یادگیری در کمینه محلی گیر نکند.

- تعداد سمبل‌های ورودی جستجوهای عضویت (MQ Symbols)

این مشخصه تعداد رشته‌های ورودی که توسط جستجوهای عضویت به کار رفته‌اند را نمایش می دهد و به نوعی نمایشگر فضای هدف می‌باشد.

- تعداد مراحل یادگیری (Rounds)

این مشخصه تعداد کل مراحل یادگیری ماشین فرضی را نمایش می‌دهد.

- مدت زمان یادگیری (Learning Time)

این مشخصه زمان کلی اجرای الگوریتم برای یادگیری ماشین را به واحد میلی ثانیه نمایش می‌دهد.

- مدت زمان یافتن مثال نقض (CE Time)

این مشخصه زمان کلی سپری شده برای یافتن مثال های نقض را به واحد میلی ثانیه نمایش می‌دهد.

در این پژوهش مشخصه اصلی مد نظر ما برای اعتبار سنجی الگوریتک تعداد بازنشانی‌های جستجوهای عضویت و معادلسازی می‌باشد. هر چقدر این دو مشخصه بیشتر کاهش یابند، بهبود کارایی الگوریتم مطلوب‌تر است.

۳.۴ مدلسازی الگوریتم‌های کلاسیک

در این پژوهش ما از سه الگوریتم کلاسیک L^*، Kearns and Vazirani و TTT استفاده کردیم. پس از پیاده‌سازی این الگوریتم‌ها و اجرای آن‌ها بر روی دو مدل خط محصولات نرم‌افزاری MP و BCS به نتایج زیر رسیدیم.

مطالعه ما نشان می‌دهد که الگوریتم‌های مبتنی بر درخت کرنز و وزیرانی و TTT بهینه‌تر از الگوریتم کلاسیک L* عمل می‌کنند و یادگیری را با تعداد کمتری از بازنشانی‌های عضویت و معادلسازی به پایان می‌رسانند (تصاویر ۱.۴ و ۲.۴)

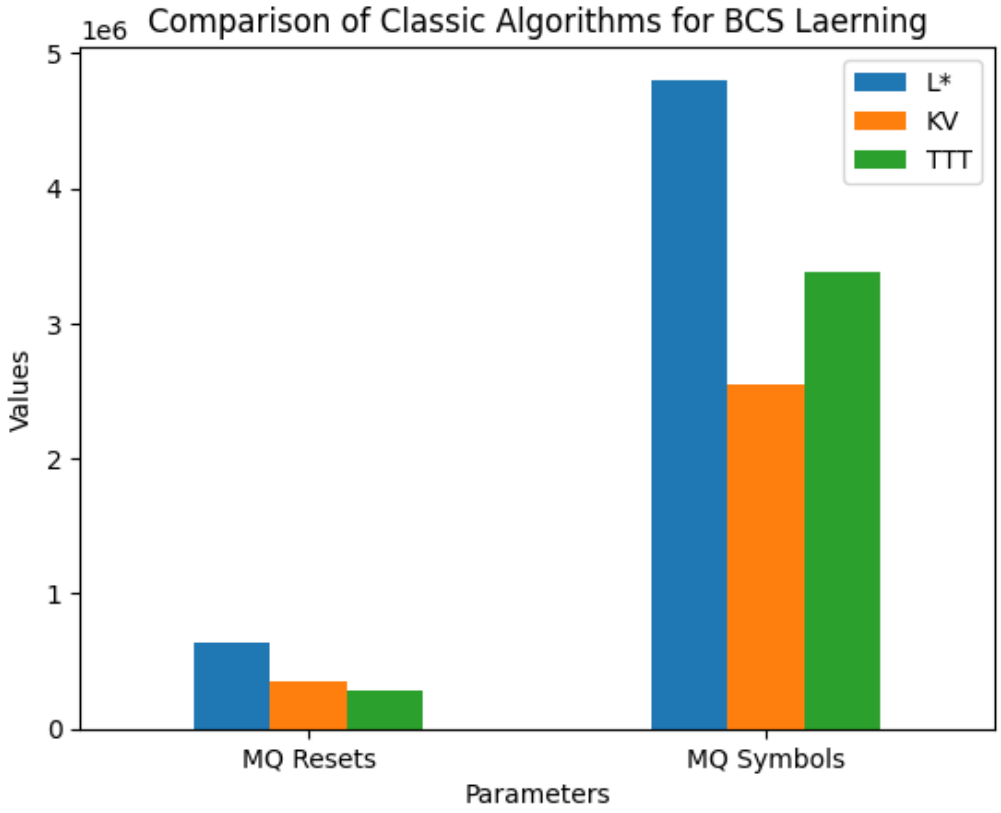
همچنین در یادگیری یک محصول مشابه الگوریتم TTT معمولا بهتر از الگوریتم Kearns and Vazirani عمل می‌کند. این موضوع به دلیل پیچیدگی کم فضای ماشین فرض محصولات است و الگوریتم TTT که به صورت تصادفی رشته‌های ورودی برای آزمایش انتخاب می‌کند در فضاهای کوچک بهتر از الگوریتم Kearns and Vazirani عمل می‌کند (جدول ۱.۴ و ۲.۴)

Adaptive KV	PL*	TTT	KV	L*	
۳۰۱۳۸	۷۶۳۷۰	۲۷۸۶۱	۳۱۲۵۹	۷۳۹۳۷	MQ Resets
۱۶۵۳۲۷	۴۱۸۴۴۰	۱۵۵۴۹۷	۱۶۷۴۷۹	۴۰۱۶۱۳	Symbols lrMQ
۲۰۸	۱۸	۲۰۸	۲۰۸	۳۰	Rounds
۵۷۷۳۲۴۷	۳۷۶۹۵۶۷	۵۸۹۰۴۰۶	۵۸۹۰۴۰۶	۴۳۵۵۶۱۹	EQ Resets
۵۸۰۳۳۸۵	۳۸۴۵۹۳۷	۵۹۱۸۲۶۷	۵۹۲۱۶۶۵	۴۴۲۹۵۵۶	Total Resets

جدول ۱.۴: نتایج اجرای الگوریتم‌های بر روی خط محصولات نرم‌افزاری Minepump

Adaptive KV	PL*	TTT	KV	L*	
۳۴۰۶۱۰	۶۷۹۶۷۲	۲۸۷۹۸۱	۳۴۴۵۴۸	۶۴۲۴۱۲	MQ Resets
۲۵۴۳۱۲۲	۵۳۲۶۶۵۴	۳۳۷۷۷۹۳	۲۵۵۲۰۲۰	۴۸۰۴۰۷۹	Symbols lrMQ
۱۷۵۳	۱۶	۱۷۵۲	۱۷۵۳	۲۲	Rounds
۱۱۰۳۱۰۶۸۸	۷۵۵۲۲۵۱۰	۱۱۰۵۳۰۶۲۲	۱۱۰۴۲۱۷۴۵	۸۲۶۹۰۴۶۷	EQ Resets
۱۱۰۶۵۱۲۹۸	۷۶۲۰۲۱۸۲	۱۱۰۸۱۸۶۰۳	۱۱۰۷۶۶۲۹۳	۸۳۳۳۲۸۷۹	Total Resets

جدول ۲.۴: نتایج اجرای الگوریتم‌های بر روی خط محصولات نرم‌افزاری Body Comfort System



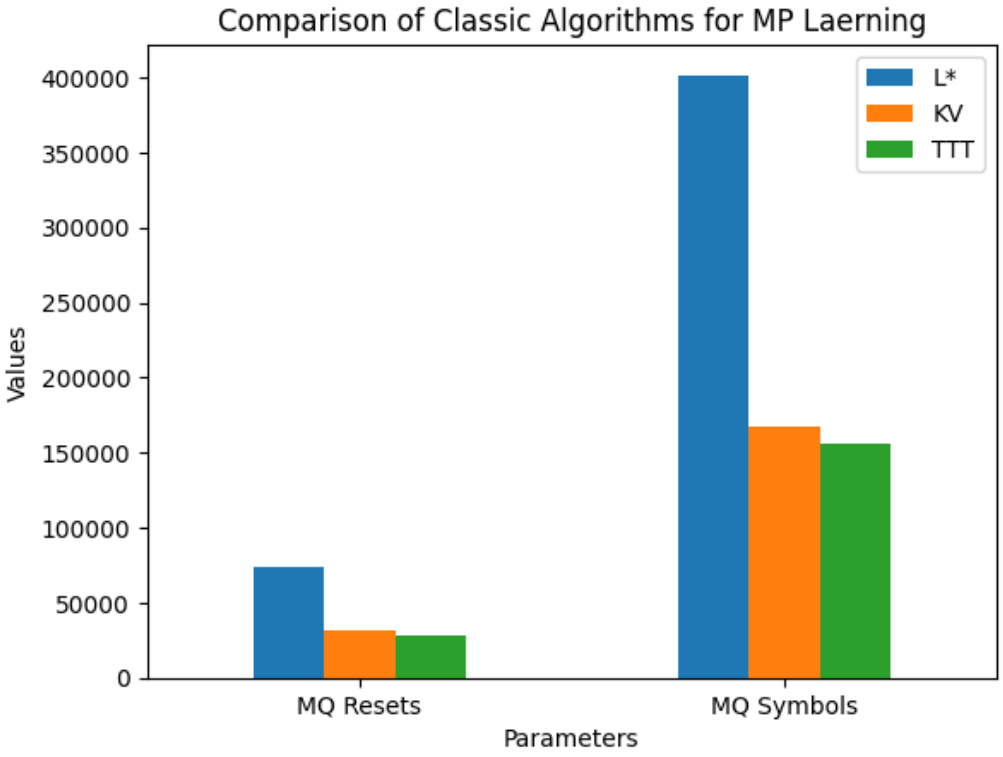
شکل ۱.۴: مقایسه‌ی الگوریتم‌های کلاسیک برای یادگیری خط محصولات نرم‌افزاری Body Comfort System

۴.۴ مدلسازی الگوریتم‌های تطبیقی

در این بخش توضیح نتایج مدلسازی الگوریتم‌های تطبیقی آورده شده است و در رابطه با این نوع الگوریتم‌ها توضیح داده می‌شود.

برای پیاده‌سازی الگوریتم کرنز و وزیرانی به صورت تطبیقی، از پیاده‌سازی توضیح داده شده در فصل قبل استفاده کردیم و آن را بر روی دو مدل خط محصولات نرم‌افزاری BCS و NP آزمودیم. برای انجام این کار به این صورت عمل شد که برای هر کدام از این خط‌های نرم‌افزاری یک ترتیب به‌خصوص درنظر گرفته شد. به این ترتیب که در یادگیری، جستجوهای عضویت مدل قبلی برای استفاده به عنوان ورودی به مدل محصول کنونی داده شد. نتایج این پیاده‌سازی به شرح زیر است:

- اجرای الگوریتم کرنز و وزیرانی تطبیقی نسبت به حالت غیرتطبیقی بهبود چشم‌گیری در یادگیری نشان

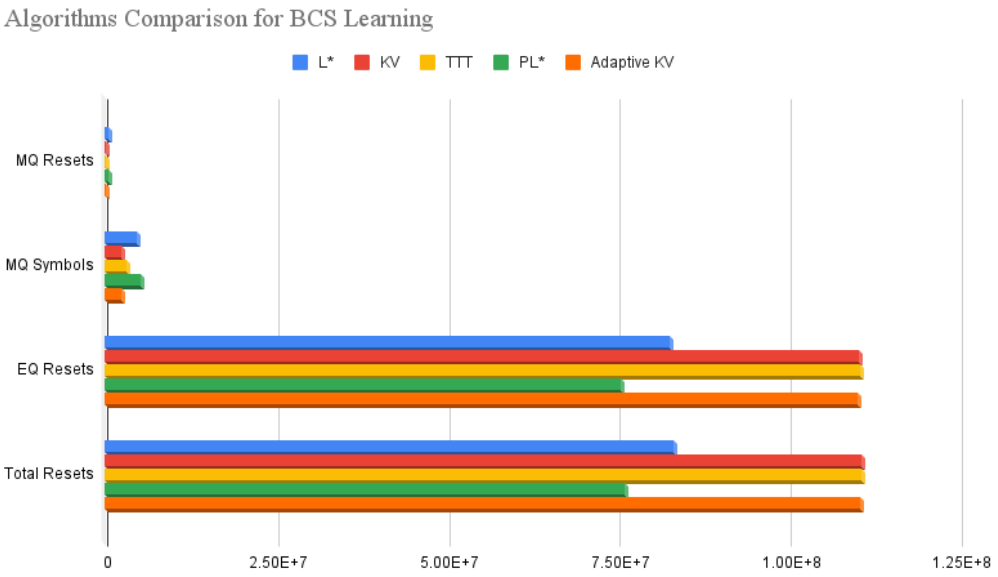


شکل ۲.۴: مقایسه‌ی الگوریتم‌های کلاسیک برای یادگیری خط محصولات نرم‌افزاری Minepump

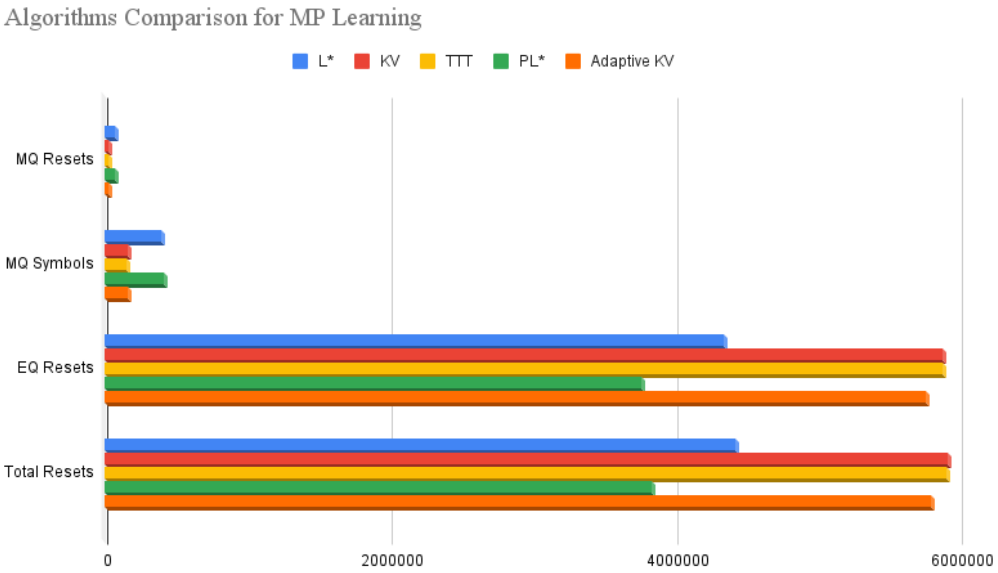
نداد. دلیل این موضوع این است که جستجوهایی که برای استفاده مجدد انتخاب شدند به صورت تصادفی و صرفا برای تطابق با الفبای مدل بعدی انتخاب شدند. این موضوع ممکن است باعث شده باشد که مدل تطبیق‌یافته نسبت به این جستجوهای خاص جهت‌گیری داشته باشد و قابلیت تعمیم‌پذیری روی داده‌های جدید را نداشته باشد.

- الگوریتم غیرتطبیقی و کلاسیک کرنز و وزیرانی که مبتنی بر درخت می‌باشد نسبت به الگوریتم کلاسیک L* در یادگیری محصولات بهتر عمل می‌کند.

جزئیات یادگیری این الگوریتم‌ها در تصاویر ۳.۴و ۴.۴ و جداول ۱.۴ و ۲.۴ قابل مشاهده است.



شکل ۳.۴: مقایسه‌ی الگوریتم‌ها برای یادگیری خط محصولات نرم‌افزاری Body Comfort System



شکل ۴.۴: مقایسه‌ی الگوریتم‌ها برای یادگیری خط محصولات نرم‌افزاری Minepump

فصل ۵

بحث و نتیجه‌گیری

۱.۵ مقدمه

در ابتدای این مقاله، به مقدمه‌ای از مدل توسعه‌ای محصولات خط نرم‌افزار و کاربرد آنها در صنعت اشاره کردیم. در ادامه چرایی انجام این پژوهش و نیاز به یادگیری تطبیقی مطرح شد. در فصل دوم پژوهش‌های پیشین و ارتباط آنها با تحقیق کنونی بررسی شدند. فصل سوم به تفصیل به توضیح مدلسازی این مسئله و روش انجام تحقیق پرداخت. در فصل چهارم نتایج و داده‌های حاصل از یادگیری الگوریتم‌های متفاوت بررسی و تحلیل و باهم مقایسه شدند. در این فصل قصد داریم تا به جمع‌بندی نتایج پیشین و کارهایی که در آینده در این زمینه قابل انجام است بپردازیم.

۲.۵ محتوا

۱.۲.۵ جمع‌بندی

به طور کلی نتایج تحقیق به شرح زیر است:

- به طور کلی الگوریتم‌های کلاسیک و تطبیقی مبتنی بر درخت مانند کرنز و وزیرانی و TTT بهینه‌تر از

الگوریتم‌های غیر درختی مانند *L و *PL عمل می‌کنند.

- الگوریتم مبتنی بر درخت کرنز و وزیرانی قابلیت یادگیری تطبیقی را دارد اما میزان بهبود آن بستگی به کیفیت و تعداد جستجوهای عضویت انتخاب‌شده برای یادگیری مجدد دارد.

۲.۲.۵ نوآوری

نوآوری اصلی این پژوهش در پیاده‌سازی الگوریتم تطبیقی مبتنی بر درخت برای یادگیری ماشین‌های حالت متناهی از جنس ماشین میلی می‌باشد. همانطور که پیشتر اشاره شده است، اکثر پژوهش‌های پیشین به یادگیری ماشین‌های قطعی متناهی پرداخته‌اند که از نظر خروجی هر حالت با ماشین میلی متفاوت است. نکته دیگر حائز اهمیت در این پژوهش استفاده از جستجوهای عضویت برای پیاده‌سازی الگوریتم تطبیقی درختی می‌باشد. در دیگر پژوهش‌هایی که به پیاده‌سازی الگوریتم تطبیقی مبتنی بر درخت پرداخته‌اند از درخت تفاوت مدل‌های یادگیری شده قبلی استفاده شده و یادگیری به صورت افزایشی مدلسازی شده است.

۳.۲.۵ پیشنهادها

محققان بعدی می‌توانند با پیاده‌سازی الگوریتم‌هایی که به پیدا کردن جستجوهای عضویتی که اطلاعات بیشتری در مورد مدل در حال یادگیری دارند، می‌پردازد، بهینگی این الگوریتم یادگیری را بهبود ببخشند. بهتر است با استفاده از یک الگوریتم مرحله به مرحله، جستجوها به صورت پویا و بر اساس حالت کنونی ماشین فرض پیدا شوند تا برای بازتعریف فرض اطلاعات بیشتری داشته باشند.

۴.۲.۵ محدودیت‌ها

در این پرژه محدودیت‌هایی داشتیم که کنترل آنها از دست ما خارج بود. از جمله این محدودیت‌ها می‌توان به موارد زیر اشاره کرد:

- ظرفیت محدود جستجوها:

کارایی الگوریتم تطبیقی به تعداد و کیفیت جستجوهای حاصل از مدل قبلی وابسته است. در نتیجه اگر

تعداد جستجوها کافی نباشد یا جستجوها نتوانند نمایانگر فضای داده‌ها باشند، کارایی الگوریتم نسبت به الگوریتم غیرتطبیقی بهبود چشمگیری نخواهد داشت.

- تعصب [1] موجود در مدل های یادگیری شده قبلی:

مدل‌های یادگیری شده قبلی ممکن است نسبت به مجموعه‌ای خاص از داده‌ها تعصب داشته باشند. این مشکل باعث می‌شود تا جستجوهای حاصل از یادگیری برای یادگیری مدل بعدی به اندازه کافی کارا نباشند.

- تطبیق بیش از حد [2] [3] و تعمیم:

الگوریتم تطبیقی ممکن است مستعد تطبیق بیش از حد جستجوهای دریافت‌شده از مدل‌های از پیش آموخته‌شده باشد، که منجر به تعمیم ضعیف به داده‌های جدید می‌شود. ممکن است ایجاد تعادل بین تطبیق مدل با جستجوها و اطمینان از تعمیم آن به داده های جدید دشوار باشد. همین مورد می‌تواند منجر به کاهش کارایی یادگیری تطبیقی شود.

[1] Bias
[2] Overfitting
[3] Generalization

کتاب‌نامه

[1] Damasceno, Carlos Diego N, Mousavi, Mohammad Reza, and da Silva Simão, Adenilso. Learning to reuse: Adaptive model learning for evolving systems. In *Integrated Formal Methods: 15th International Conference, IFM 2019, Bergen, Norway, December 2–6, 2019, Proceedings*, pages 138–156. Springer, 2019.

[2] Ferreira, Tiago, van Heerdt, Gerco, and Silva, Alexandra. Tree-based adaptive model learning. pages 164–179, 2022.

[3] Kearns, Michael J and Vazirani, Umesh. *An introduction to computational learning theory*. MIT press, 1994.

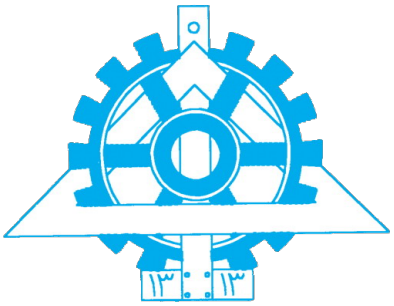
[4] Tavassoli, Shaghayegh, Damasceno, Carlos Diego N, Khosravi, Ramtin, and Mousavi, Mohammad Reza. Adaptive behavioral model learning for software product lines. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference*, volume A (SPLC '22), pages 142–153, Graz, Austria, September 12–16 2022.

[5] Tavassoli, Shaghayegh, Damasceno, Carlos Diego N, Mousavi, Mohammad Reza, and Khosravi, Ramtin. A benchmark for active learning of variability-intensive systems. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference*, volume A (SPLC '22), pages 245–249, Graz, Austria, September 12–16 2022.

Abstract

Adaptive learning is a new method for analyzing the performance of Software Product Lines (SPLs) and model-based testing. In this type of learning, the focus is on creating a behavioral model of the SPL. Due to commonalities between different products of an SPL, it is possible to use the previously learned models to improve efficiency and learning performance. Behavioral models often do not exist in the real world or are obsolete and therefore require further analysis from previous implementations. In these cases, it is preferable to use adaptive model learning. Many variables cause challenges in this type of learning, the most important of which is the number of queries performed, which in many cases makes learning impossible. For this reason, it is necessary to use adaptive learning algorithms instead of previous space and time-consuming classic algorithms. In this project, we intended to measure the improvement of the learning performance by applying an adaptive model based on the famous Kearns and Vazirani tree-based algorithm. Two important components in learning are membership queries and equivalence queries. The main criteria for measuring performance in this research are the number of learning rounds , equivalence queries, resets and the total number of input symbols In this project, after the initial implementation of the non-adaptive algorithm, we obtained the discrimination tree used in the learning. Then using the algorithm defined in this article, we extracted a file of the queries' appropriate inputs and outputs stored in the discrimination tree. In the next step, the adaptive learning algorithm was implemented, which by receiving the file obtained from learning the previous product, builds a tree based on this information and uses it to learn the new product. In this way, by reusing the previous product queries, the number of subsequent product queries will be reduced, which will be an important step in processing and analyzing SPLs.

Keywords Tree-based Adaptive Model Learning, Behavioral Model, Software Product Lines, Kearns and Vazirani Algorithm



University of Tehran
College of Engineering
Faculty of Electrical and Computer Engineering
Software Engineering

Tree-based adaptive model learning of software product lines

Bachelor of Science Thesis in Computer Engineering

By:

Bahar Emami Afshar

Supervisor:

Dr. Ramtin Khosravi

June 2023