دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی ارشد
مهندسی رایانش امن

بهبود کارایی روش‌های تشخیص برنامک‌های اندرویدی بازبسته‌بندی شده

نگارش

مجتبی موذن

استاد راهنما

دکتر مرتضی امینی

بهمن ۱۴۰۱

به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی ارشد

این پایان‌نامه به عنوان تحقق بخشی از شرایط دریافت درجه کارشناسی ارشد است.

عنوان: بهبود کارایی روش‌های تشخیص برنامک‌های اندرویدی بازبسته‌بندی شده

نگارش: مجتبی موذن

کمیته ممتحنین

استاد راهنما: دکتر مرتضی امینی امضاء:

استاد مشاور: استاد مشاور امضاء:

استاد مدعو: استاد ممتحن امضاء:

تاریخ:

سپاس

از استاد بزرگوارم که با کمک‌ها و راهنمایی‌های بی‌دریغشان، مرا در به سرانجام رساندن این پایان‌نامه یاری داده‌اند، تشکر و قدردانی می‌کنم. همچنین از همکاران عزیزی که با راهنمایی‌های خود در بهبود نگارش این نوشتار سهیم بوده‌اند، صمیمانه سپاسگزارم.

چکیده

با گسترش روزافزون استفاده از برنامک‌های اندرویدی در سالیان اخیر حملات موجود بر روی این سیستم‌عامل با افزایش قابل توجهی همراه بوده‌است. متن‌باز بودن برنامک‌های اندرویدی و در نتیجه، دسترسی به کد منبع این دسته از برنامک‌ها، در کنار افزایش حملات بر روی آن‌ها ، لزوم توجه به مقابله با حملات مطروحه در این زمینه را افزایش داده‌است.حملات بازبسته‌بندی روی برنامک‌های اندرویدی، نوعی از حملات هستند که در آن مهاجم، پس از دسترسی به کد منبع برنامک و کپی‌کردن آن و یا ایجاد تغییراتی که مدنظر مهاجم است، مجدداً آن‌را بازبسته‌بندی می‌کند.تغییر کدهای برنامک، اهداف متفاوتی نظیر تغییر کتابخانه‌های تبلیغاتی، نقض امنیت کاربر و یا ضربه به شرکت‌های تولید برنامک از تغییر گسترش برنامک‌های جعلی را دنبال می‌کند. بازبسته‌بندی برنامک‌های اندرویدی علاوه بر ماهیت تهدید کاربران و شرکت‌ها، ماهیتی پیشگیرانه نیز دارد. در این حالت توسعه‌دهندگان نرم‌افزار از طریق ایجاد مبهم‌نگاری در برنامک‌های اندرویدی، سعی در پیشگیری از بازبسته‌بندی به وسیله‌ی مهاجمان دارند. تشخیص بازبسته‌بندی در برنامک‌های اندرویدی از آن‌جهت دارای اهمیت است که هم کاربران و هم شرکت‌های توسعه‌دهنده، می‌توانند از این موضوع ذی‌نفع،باشند. تشخیص برنامک‌های بازبسته‌بندی شده، به جهت چالش‌های پیش‌رو، نظیر مبهم‌نگاری کدهای برنامک جعلی به دست مهاجم و همچنین تشخیص و جداسازی صحیح کدهای کتابخانه‌ای مسئله‌ای چالشی محسوب می‌شود. پژوهش‌های اخیر در این زمینه به صورت کلی، از روش‌های تشخیص مبتنی بر شباهت‌سنجی کدهای برنامک و یا طبقه‌بندی برنامک‌های موجود استفاده کرده‌اند. از طرفی برقراری حدواسطی میان سرعت و دقت در تشخیص برنامک‌های جعلی، چالشی است که استفاده از این دست پژوهش‌ها را در یک محیط صنعتی ناممکن ساخته‌است. در این پژوهش پس‌از استخراج کدهای برنامک به وسیله‌ی چارچوب سوت و ابزارهای دیس‌اسمبل، در یک روش دو مرحله‌ای کدهای برنامک‌های موجود با یکدیگر مقایسه می‌شود. پس از دیس‌اسمبل کدهای هر برنامک، در طی یک فرایند طبقه‌بندی مبتنی بر ویژگی‌های انتزاعی و دیداری، برنامک‌های کاندید برای هر برنامک مبدا استخراج می‌شود. سپس برای هر کلاس برنامک اندرویدی، امضایی متشکل از مهم‌تری ویژگی‌های کدپایه از آن استخراج و پس از انجام مقایسه با کلاس‌های کتابخانه‌های اندرویدی موجود در مخزن، کتابخانه‌های اندرویدی حذف می‌شوند و در نهایت با مقایسه‌ی کدهای اصلی، برنامک بازبسته‌بندی شده مشخص می‌شود. در قسمت آزمون روش پیشنهادی در این پژوهش، توانستیم روش موجود در این زمینه‌را با بهبود امضای تولیدشده از هر برنامک و اضافه‌شدن مرحله‌ی پیش‌پردازش، سرعت تشخیص را ۴ برابر افزایش داده و در عین‌حال دقت روش موجود را نیز حفظ کنیم.

کلیدواژه‌ها: پایان‌نامه، حروف‌چینی، قالب، زی‌پرشین

فهرست مطالب

فهرست جدول‌ها

فهرست شکل‌ها

فصل ۱

مقدمه

سیستم‌عامل اندروید به دلیل سهولت در توسعه توسط توسعه‌دهندگان موبایلی و در نتیجه فراوانی استفاده از آن در تلفن‌های همراه، تلوزیون‌های هوشمند و دیگر دستگاه‌های موجود، حجم بالایی از بازار سیستم‌عامل‌های موبایلی را به خود اختصاص داده‌است. بر طبق گزارش پایگاه استاتیستا [۱] سیستم‌عامل اندروید سهمی معادل ۷۱ درصدی از سیستم‌عامل‌های موبایلی را در سه ماهه‌ی پایانی سال ۲۰۲۲ به خود اختصاص داده‌است. در سال‌های اخیر به دلیل گسترش استفاده از این پلتفرم، فروشگاه‌های اندرویدی زیادی به جهت ارائه‌ی خدمات به کاربران به وجود آمده‌است. برخی از فروشگاه‌های رسمی مانند فروشگاه اندرویدی گوگل، از ابزارهایی نظیر پلی پروتکت[۲] برای بررسی برنامک‌های اندرویدی موجود در فروشگاّه استفاده می‌کنند. علاوه بر این، در سال‌های اخیر فروشگآه‌های متعدد رایگانه به وجود آمده‌اند که صرفا برنامک‌های اندرویدی موجود در سطح وب را غربان و آن را به کاربران ارائه می‌دهند. فروشگاه‌های رایگان غالباً ابزارهای مشخصی را برای حفظ امنیت کاربران استفاده نمی‌کنند و امنیت کاربران این دسته از فروشگاه‌های اندرویدی، ، همواره تهدید می‌شود یکی از راه‌های مورد استفاده توسط مهاجمان برای وارد ساختن بدافزار به تلفن‌های همراه، بازبسته‌بندی نرم‌افزار است. مطابق تعریف، بازبسته‌بندی شامل دانلود یک برنامک، دسترسی به محتوای کدهای برنامک اصلی از طریق روش‌های مهندسی معکوس و در نهایت بازبسته‌بندی به همراه تغییر و یا بدون تغییردادن کدهای برنامک اصلی است. زبان اصلی توسعه در برنامک‌های اندرویدی، زبان جاوا می‌باشد که یک زبان سطح بالا محسوب می‌شود. در طی فرآیند کامپایل برنامک‌های اندرویدی، مجموعه‌ی کدهای منبع در طی فرایندی به بایت‌کدهای دالویک تبدیل می‌شوند و در ادامه ماشین مجازی جاوا، بایت‌کدها را برروی ماشین مقصد اجرا می‌کند[۳]. فهم و در نتیجه مهندسی معکوس زبان میانی دالویک بایت‌کدها آسان است و به همین علت موجب سهولت در بازبسته‌بندی برنامک‌های اندرویدی می‌شود.

به طور کلی بازبسته‌بندی را می‌توان از دو جهت مورد بررسی قرار داد، از دید توسعه‌دهندگان، بازبسته‌بندی

شامل فرآیندی است که توسعه‌دهنده با انجام مبهم‌نگاری در برنامک مورد توسعه، فهم بدنه‌ی اصلی برنامک را برای مهاجم سخت می‌کند. از این دید، بازبسته‌بندی یک روش تدافعی تلقی می‌شود تا مهاجم پس از دسترسی به کد برنامک اصلی، نتواند بدنه‌ی برنامک اصلی برنامک را شناسایی و در نتیجه آن‌را تغییر دهد. از جهت دیگر، بازبسته‌بندی توسط فردی که برنامک متعلق به او نیست یک عمل تهاجمی محسوب می‌شود. در این حالت، مهاجم پس از دسترسی به کد برنامک‌اصلی، بسته به هدف او، برنامک را مجددا بازبسته‌بندی می‌کند و آن را در فروشگاه‌های اندرویدی خصوصا فروشگاه‌هایی که نظارت کمتری بر روی آن‌ها وجود دارد منتشر می‌کند. در این حالت مهاجم به جهت اهدافی متفاوتی نظیر تغییر کدهای تبلیغاتی در برنامک اصلی، تغییر درگاه‌های پرداخت و یا بازپخش بدافزار، اقدام به بازبسته‌بندی می‌کند. بازبسته‌بندی یکی از راه‌های محبوب مهاجمان برای انتقال بدافزارهای توسعه‌داده‌شده به تلفن همراه قربانی است[۴]. مطابق پژوهش آقای ژو و همکاران[۵] حدود۸۵ درصد بدافزارهای موجود، از طریق بازبسته‌بندی منتشر می‌شوند. همانطور که گفته شد، برخی فروشگآهای اندرویدی نظیر گوگل، سازوکار مشخصی را برای تشخیص بازبسته‌بندی ارائه‌داده‌اند اما بسیاری از فروشکاه‌های اندرویدی فعال و پربازدید، خصوصا فروشگاه‌های رایگان، یا از هیچ ابزاری استفاده نمی‌کنند و یا در صورت توسعه‌ی نرم‌افزار بومی خود برای شناسایی برنامک‌های بازبسته‌بندی شده، مشخصات و یا دقت آن را گزارش نکرده‌اند.

همانطور که اشاره‌شد، محبوبیت و در نهایت استفاده‌ی زیاد برنامک‌های اندرویدی و همچنین نظارت کم در فروشگاه‌های اندرویدی، بازبسته‌بندی، یک روش پر استفاده به جهت انتقال بدافزار به تلفن همراه کاربران است. آقای خانمحمدی و همکاران[۶]، پس از بررسی پایگاه‌داده‌ای برنامک‌های اندرویدی اندروزو، دریافتند که ٪۵۲/۲۲ از برنامک‌های موجود در این مخزن توسط ویروس‌توتال، بدافزار شناسایی شده‌اند. ویروس‌توتال، ابزاری متشکل از ۳۰ ضدبدافزار برای بررسی یک برنامک اندرویدی اندرویدی است. مطابق این پژوهش، ٪۷۷/۸۴ از برنامک‌های این مخزن‌داده که بازبسته‌بندی شده‌اند، دارای نوعی از بدافزار ضدتبلیغاتی بوده‌اند که موجب می‌شود تبلیغات موجود در برنامک تغیر و اهداف مالی و امنیتی کاربران و توسعه‌دهندگان مخدوش شود . علاوه بر این، مطابق پژوهشی که توسط ویداس وهمکاران[۷] انجام شده‌است، پس از پیاده‌سازی ۷ روش پربازدید به جهت تشخیص بازبسته‌بندی، در بهترین حالت، روش‌های موجود قادر به تشخیص ٪۷۲/۲۲ از برنامک‌های بازبسته‌بندی شده‌ی سه فروشگاه مطرح اندرویدی است. بنابراین مشخص است که تشخیص برنامک‌های بازبسته‌بندی شده، به چه میزان می‌تواند اهداف مالی و امنیتی توسعه‌دهندگان و کاربران برنامک‌ها را ارضا کند. در سال‌های اخیر ارائه‌ی یک راهکار پرسرعت به همراه دقت مناسب، همواره یکی از دغدغه‌های مهم پژوهش‌کنندگان در این زمینه بوده‌است.

همانطور که گفته‌شد، بازبسته‌بندی برنامک‌های اندرویدی از دو دیدگاه تهاجمی و تدافعی قابل بررسی است. در حالتی که کاربر متقلب، برنامک اندرویدی اصلی را دجار تغییراتی می‌کند و آن را در اختیار عموم قرار می‌دهد، تشخیص بازبسته‌بندی، با استفاده از مقایسه‌ی برنامک اصلی و برنامک جعلی صورت می‌گیرد.

تشخیص بازبسته‌بندی در این حالت را می‌تواند در حالت کلی به دو طبقه تقسیم کرد. در حالت اول توسعه‌دهنده روش خود را مبتنی بر تحلیل برنامک مبدا و مقصد پیاده‌سازی می‌کند. عمده‌ی روش‌های موجود در این طبقه مبتنی بر تحلیل ایستای جفت برنامک‌ها است و استفاده از تحلیل پویا به جهت سرعت پایین آن، محبوبیت فراوانی ندارد و بیشتر از تحلیل ایستای برنامک‌های اندرویدی استفاده می‌شود[۸]. در سمت دیگر طبقه‌بندی برنامک‌های اندرویدی وجود دارد.روش‌های موجود در این دسته، عمدتا سرعت بالایی دارند اما در تشخیص جفت بازبسته‌بندی شده دقت پایینی را ارائه می‌دهند.

برنامک‌های اندرویدی متشکل از دو قسمت اصلی کدهای برنامک و منابع آن هستند. کدهای برنامک، منطق برنامک را تشکیل می‌دهند و رفتار برنامک با توجه به این قسمت مشخص می‌شود. از طرفی منابع برنامک، ظاهر آن را تشکیل می‌دهند. روش‌های مبتنی بر تحلیل برنامک و یا طبقه‌بندی آن، عمدتاً از ویژگی‌های موجود در منابع و یا کد استفاده می‌کنند. مهاجم در حالاتی که می‌خواهد از محبوبیت برنامک مبدا استفاده کند، سعی در یکسان‌سازی ظاهر برنامک‌های مبدا و مقصد دارد به همین جهت از منابع برنامک مبدا استفاده می‌کند و منطق برنامک را مطابق با اهداف خود تغییر می‌دهد. در حالتی دیگر، متقلب سعی می‌کند که با استفاده از تغییر منابع برنامک و تولید یک برنامک تقلبی و گاهاً بدون هیچ تغییری در کد برنامک، ادعای توسعه‌ی یک برنامک جدید را اثبات کند. لازم به ذکر است استفاده از ویژگی‌های کدپایه و منبع‌پایه، به وفور در پژوهش‌های سال‌های اخیر یافت می‌شود که هر کدام معایب و مزایای خود را دارد.

در روش‌های مبتنی بر طبقه‌بندی عمدتا تعریف تشخیص بازبسته‌بندی محدود به تشخیص دسته‌ی مشکوک و یا دسته‌ای از برنامک‌ها است که احتمال بازبسته‌بندی بودن جفت‌های داخل این دسته، بیش از سایر دسته‌ها است. بنابراین تشخیص بازبسته‌بندی در این روش‌ها، محدود به تشخیص طبقه‌ی برنامک ورودی می‌باشد و جفت بازبسته‌بندی شده مشخص نمی‌شود. از طرفی در روش‌های مبتنی بر تحلیل ایستا، بررسی دوبه‌دوی برنامک‌های ورودی و مجموعه‌داده مدنظر است. در این روش‌ها تعریف تشخیص بازبسته‌بندی گسترش یافته و یافتن جفت بازبسته‌بندی به صورت مشخیص، هدف پژوهش می‌شود. تغییر منابع برنامک و همچنین مبهم‌نگاری در برنامک بازبسته‌بندی شده، دوجالش مهم در راستای تشخیص بازبسته‌بندی است. متقلب پس از بازبسته‌بندی برنامک، با استفاده از مبهم نگاری سعی می‌کند تغیرات خود و شباهت ساختار منطقی برنامک تقلبی با برنامک اصلی را پنهان کند. به همین جهت، تشخیص بازبسته‌بندی نیازمند ویژگی‌هایی است که مقاومت بالایی مقابل مبهم‌نگاری داشته‌باشد بدین معنا که تغییر و ایجاد ابهام در کد، به راحتی در این ویژگی‌ها قابل انجام نباشد.

در هنگار کامپایل برنامک‌های اندرویدی، کتابخانه‌هایی که در برنامک مورد استفاده قرار گرفته‌اند به همراه کد مورد توسعه، کامپایل شده و دالویک بایت‌کدهای آن در کنار برنامک قرار می‌گیرد. بر اساس پژوهش آقای زیانگ و همکاران [۹] ٪۵۷ از کدهای برنامک‌های مورد بررسی در این پژوهش، شامل کدهای کتابخانه‌ای بودند که دجار مبهم‌نگاری نشده‌اند. بنابراین تشخیص کدهای بازبسته‌بندی شده بدون تشخیص درست

و دقیق و جداسازی کدهای کتابخانه‌ای امکان‌پذیر نیست و می‌تواند نتایج منفی غلط و مثبت غلط را کاهش دهد. به صورت کلی دو روش برای تشخیص کدهای کتابخانه‌ای استفاده می‌شود، روش مبتنی بر لیست سفید و یا روش تشخیص مبتنی بر شباهت سنجی. در روش لیست سفید، لیسیتی از مشهور ترین کتابخانه‌های موجود در مخازن کتابخانه‌های اندرویدی نظیر ماون را جمع آوری می‌شود و با استفاده از نام کلاس‌ها و بسته‌های موجود کلاس‌های کتابخانه ای تشخیص‌داده می‌شود. مشخص است که این روش مقاومت بسیار کمی مقابل ساده‌ترین روش‌های مبهم‌نگاری در کتابخانه‌های اندرویدی دارد. در حالت دیگر از روش‌های مبتنی بر شباهت سنجی برای تشخیص کدهای کتابخانه‌ای استفاده می‌شود که در این روش، تحلیل ایستا روی کدهای برنامک مبدا و مخزن کتابخانه‌های اندرویدی صورت می‌گیرد و در نهایت با یکدیگر مقایسه می‌شوند. مشخص است که روش‌های مبتنی بر شباهت‌سنجی از دقت بیشتری برخوردار هستند و تمایز بهتری میان کدهای کتابخانه‌ای و کدهای اصلی قرار می‌دهند اما اینگونه روش‌ها سرعت پایینی دارند.

پژوهش‌های ارائه‌شده در زمینه‌ی تشخیص برنامک‌های بازبسته‌بندی شده در سال‌های اخیر، عمدتاً در تلاش برای بهبود دقت و سرعت روش‌های پیشین بوده‌اند.مبهم‌نگاری باعث می‌شود که دقت روش‌های تشخیص مبتنی بر تحلیل ایستا و شباهت سنجی پایین بیاید و استفاده از ویژگی‌هایی را که مقاومت بالایی مقابل مبهم‌نگاری داشته باشند را واجب کند. از طرفی استفاده از ویزگی‌های مقاوم به مبهم‌نگاری، می‌تواند سرعت تشخیص را بسیار پایین آورده تا حدی که عملا استفاده از این روش‌ها در یک محیط صنعتی را غیر ممکن سازد. در این پژوهش ما با استفاده از ترکیب روش‌های تحلیل ایستا و طبقه‌بندی منابع، به همراه شباهت‌سنجی، روشی را ارائه کرده‌ایم که در حالی که مقاومت بالایی نسبت به مبهم‌نگاری داشته‌باشد، سرعت روش‌های پیشین را نیز افزایش دهد. در این پژوهش به عنوان پیش‌پردازش، از یک طبقه‌بند نزدیک‌ترین همسایه برای کاهش فضای مقایسه‌ی دودویی و با استفاده از ویژگی‌های مبتنی بر منبع، استفاده شده‌است. با کاهش فضای مقایسه‌ی دودویی و طبقه‌بندی برنامک‌های مشکوک در یک دسته، مقایسه‌ی برنامک‌های موجود در آن دسته آغاز می‌شود. مقایسه‌ی دودویی در هر دسته مبتنی بر تحلیل ایستا و شباهت‌سنجی کدهای برنامک انجام می‌شود. ابتدا ویژگی‌هایی از هر کلاس و متد در بسته‌های برنامک استخراج شده و امضای هر کلاس ساخته می‌شود به طوری که امضای هر کلاس منحصر به فرد و تا حد امکان مختص همان کلاس باشد. نوآوری روش مطروحه، ترکیب روش‌های مبتنی بر طبقه‌بندی و روش‌های مبتنی بر تحلیل می‌باشد که در نهایت منجر به افزایش سرعت و در عین حال دقت خوب در تشخیص برنامک‌های بازبسته‌بندی شده‌است. حذف کدهای کتابخانه‌ای با استفاده از روشی مبتنی بر مقایسه‌ی کدهای موجود در مخزن کتابخانه‌ها و کلاس‌های برنامک انجام می‌شود. مخزن کتابخانه‌ها متشکل از ۴۵۳ کتابخانه‌ی اندرویدی جمع‌آوری شده از مخزن ماون مب‌باشد. در نهایت پس از تشخیص کلاس‌های کتابخانه‌های اندرویدی و حذف آن‌ها از کد برنامک، کدهای مورد توسعه به عنوان ورودی برای

مقایسه‌ی دودویی مورد تحلیلی قرار می‌گیرند.

در ادامه‌ی این نگارش، در فصل ۲ به بررسی و تعریف مفاهیم اولیه‌ی مورد نیاز در این پژوهش می‌پردازیم. در فصل ۳ به تعریف مسئله می‌پردازیم و همچنین مروری از کارهای پیشین را خواهیم داشت. در ادامه و در فصل ۴ روش مورد استفاده در این پژوهش، شرح داده خواهدشد و در فصل ۵ مقایسه و ارزیابی روش پیشنهادی خود را ارائه می‌دهیم. در نهایت و در فصل ۶ ضمن جمع‌بندی این گزارش علمی، به بررسی نقاط ضعف و قوت این پژوهش و همچنین ارائه‌ی پیشنهاداتی جهت بهبود این پژوهش می‌پردازیم.

فصل ۲

مفاهیم اولیه

دومین فصل پایان‌نامه به طور معمول به معرفی مفاهیمی می‌پردازد که در پایان‌نامه مورد استفاده قرار می‌گیرند. در این فصل به عنوان یک نمونه، نکات کلی در خصوص نحوه‌ی نگارش پایان‌نامه و نیز برخی نکات نگارشی به اختصار توضیح داده می‌شوند.

۲ـ۱ نحوه‌ی نگارش

۲ـ۱ـ۱ پرونده‌ها

پرونده‌ی اصلی پایان‌نامه در قالب استاندارد[۱] `thesis.tex` نام دارد. به ازای هر فصل از پایان‌نامه، یک پرونده در شاخه‌ی `chapters` ایجاد نموده و نام آن را در `thesis.tex` (در قسمت فصل‌ها) درج نمایید. برای مشاهده‌ی خروجی، پرونده‌ی `thesis.tex` را با زی‌لاتک کامپایل کنید. مشخصات اصلی پایان‌نامه را می‌توانید در پرونده‌ی `front/info.tex` ویرایش کنید.

۲ـ۱ـ۲ عبارات ریاضی

برای درج عبارات ریاضی در داخل متن از $...$ و برای درج عبارات ریاضی در یک خط مجزا از $$...$$ یا محیط equation استفاده کنید. برای مثال عبارت $۲x + ۳y$ در داخل متن و عبارت زیر

$$\sum_{k=۰}^{n} \binom{n}{k} = ۲^n \qquad (۱ـ۲)$$

[۱] قالب استاندارد از گیت‌هاب به نشانی github.com/zarrabi/thesis-template قابل دریافت است.

در یک خط مجزا درج شده است. دقت کنید که تمامی عبارات ریاضی، از جمله متغیرهای تک‌حرفی مانند x و y باید در محیط ریاضی یعنی محصور بین دو علامت $ باشند.

۲ـ۱ـ۳ علائم ریاضی پرکاربرد

برخی علائم ریاضی پرکاربرد در زیر فهرست شده‌اند. برای مشاهده‌ی دستور معادل پرونده‌ی منبع را ببینید.

- مجموعه‌های اعداد: $\mathbb{N}, \mathbb{Z}, \mathbb{Z}^+, \mathbb{Q}, \mathbb{R}, \mathbb{C}$
- مجموعه: $\{۱, ۲, ۳\}$
- دنباله: $\langle ۱, ۲, ۳ \rangle$
- سقف و کف: $\lceil x \rceil, \lfloor x \rfloor$
- اندازه و متمم: $|A|, \overline{A}$
- همنهشتی: $a \overset{n}{\equiv} ۱$ یا (پیمانه‌ی n) $a \equiv ۱$
- ضرب و تقسیم: $\times, \cdot, \div$
- سه‌نقطه: $۱, ۲, \ldots, n$
- کسر و ترکیب: $\frac{n}{k}, \binom{n}{k}$
- اجتماع و اشتراک: $A \cup (B \cap C)$
- عملگرهای منطقی: $\neg p \vee (q \wedge r)$
- پیکان‌ها: $\to, \Rightarrow, \leftarrow, \Leftarrow, \leftrightarrow, \Leftrightarrow$
- عملگرهای مقایسه‌ای: $\neq, \leqslant, \nleqslant, \geqslant, \ngeqslant$
- عملگرهای مجموعه‌ای: $\in, \notin, \setminus, \subset, \subseteq, \subsetneq, \supset, \supseteq, \supsetneq$
- جمع و ضرب چندتایی: $\sum_{i=۱}^{n} a_i, \prod_{i=۱}^{n} a_i$
- اجتماع و اشتراک چندتایی: $\bigcup_{i=۱}^{n} A_i, \bigcap_{i=۱}^{n} A_i$
- برخی نمادها: $\infty, \emptyset, \forall, \exists, \triangle, \angle, \ell, \equiv, \therefore$

۲ـ۱ـ۴ لیست‌ها

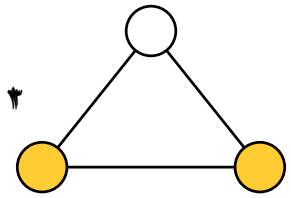
برای ایجاد یک لیست می‌توانید از محیط‌های «فقرات» و «شمارش» همانند زیر استفاده کنید.

- مورد اول
- مورد دوم
- مورد سوم

۱. مورد اول
۲. مورد دوم
۳. مورد سوم

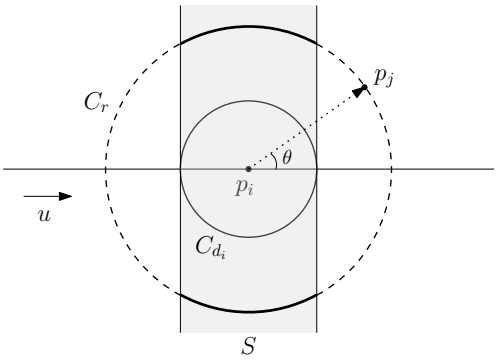
۲ـ۱ـ۵ درج شکل

یکی از روش‌های مناسب برای ایجاد شکل استفاده از نرم‌افزار LaTeX Draw و سپس درج خروجی آن به صورت یک فایل tex درون متن با استفاده از دستور fig یا centerfig است. شکل ۲ـ۱ نمونه‌ای از اشکال ایجادشده با این ابزار را نشان می‌دهد.



شکل ۲ـ۱: یک گراف و پوشش رأسی آن

همچنین می‌توانید با استفاده از نرم‌افزار Ipe شکل‌های خود را مستقیما به صورت pdf ایجاد نموده و آن‌ها را با دستورات img یا centerimg درون متن درج کنید. برای نمونه، شکل ۲ـ۲ را ببینید.



شکل ۲ـ۲: نمونه شکل ایجادشده توسط نرم‌افزار Ipe

۲ـ۱ـ۶ درج جدول

برای درج جدول می‌توانید با استفاده از دستور «جدول» جدول را ایجاد کرده و سپس با دستور «لوح» آن را درون متن درج کنید. برای نمونه جدول ۲ـ۱ را ببینید.

جدول ۲ـ۱: عملگرهای مقایسه‌ای

عملگر	عنوان
<	کوچک‌تر
>	بزرگ‌تر
==	مساوی
<>	نامساوی

۲ـ۱ـ۷ درج الگوریتم

برای درج الگوریتم می‌توانید از محیط «الگوریتم» استفاده کنید. یک نمونه در الگوریتم ۱ آمده است.

الگوریتم ۱ پوشش رأسی حریصانه

ورودی: گراف $G = (V, E)$

خروجی: یک پوشش رأسی از G

۱: قرار بده $C = \emptyset$

۲: تا وقتی E تهی نیست:

۳: یال دل‌خواه $uv \in E$ را انتخاب کن

۴: رأس‌های u و v را به C اضافه کن

۵: تمام یال‌های واقع بر u یا v را از E حذف کن

۶: C را برگردان

۲ـ۱ـ۸ محیط‌های ویژه

برای درج مثال‌ها، قضیه‌ها، لم‌ها و نتیجه‌ها به ترتیب از محیط‌های «مثال»، «قضیه»، «لم» و «نتیجه» استفاده کنید. برای درج اثبات قضیه‌ها و لم‌ها از محیط «اثبات» استفاده کنید.

تعریف‌های داخل متن را با استفاده از دستور «مهم» به صورت **تیره** نشان دهید. تعریف‌های پایه‌ای‌تر را درون محیط «تعریف» قرار دهید.

تعریف ۲ـ۱ (اصل لانه‌کبوتری) *اگر $n+1$ کبوتر یا بیش‌تر درون n لانه قرار گیرند، آن‌گاه لانه‌ای وجود دارد که شامل حداقل دو کبوتر است.*

۲ـ۲ برخی نکات نگارشی

این فصل حاوی برخی نکات ابتدایی ولی بسیار مهم در نگارش متون فارسی است. نکات گردآوری‌شده در این فصل به هیچ وجه کامل نیست، ولی دربردارنده‌ی حداقل مواردی است که رعایت آن‌ها در نگارش پایان‌نامه ضروری به نظر می‌رسد.

۲ـ۲ـ۱ فاصله‌گذاری

۱. علائم سجاوندی مانند نقطه، ویرگول، دونقطه، نقطه‌ویرگول، علامت سؤال و علامت تعجب بدون فاصله از کلمه‌ی پیشین خود نوشته می‌شوند، ولی بعد از آن‌ها باید یک فاصله قرار گیرد. مانند: من، تو، او.

۲. علامت‌های پرانتز، آکولاد، کروشه، نقل قول و نظایر آن‌ها بدون فاصله با عبارات داخل خود نوشته می‌شوند، ولی با عبارات اطراف خود یک فاصله دارند. مانند: (این عبارت) یا {آن عبارت}.

۳. دو کلمه‌ی متوالی در یک جمله همواره با یک فاصله از هم جدا می‌شوند، ولی اجزای یک کلمه‌ی مرکب باید با نیم‌فاصله[۲] از هم جدا شوند. مانند: کتاب درس، محبت‌آمیز، دوبخشی.

۴. اجزای فعل‌های مرکب با فاصله از یک‌دیگر نوشته می‌شوند، مانند: تحریر کردن، به سر آمدن.

۲ـ۲ـ۲ شکل حروف

۱. در متون فارسی به جای حروف «ك» و «ي» عربی باید از حروف «ک» و «ی» فارسی استفاده شود. همچنین به جای اعداد عربی مانند ٥ و ٦ باید از اعداد فارسی مانند ۵ و ۶ استفاده نمود. برای این

[۲] «نیم‌فاصله» فاصله‌ای مجازی است که در عین جدا کردن اجزای یک کلمه‌ی مرکب از یک‌دیگر، آن‌ها را نزدیک به هم نگه می‌دارد. معمولاً برای تولید این نوع فاصله در صفحه‌کلیدهای استاندارد از ترکیب Shift+Space استفاده می‌شود.

کار، توصیه می‌شود صفحه‌کلید فارسی استاندارد[۳] را بر روی سیستم خود نصب کنید.

۲. عبارات نقل‌قول‌شده یا مؤکد باید درون علامت نقل قولِ «» قرار گیرند، نه "". مانند: «کشور ایران».

۳. کسره‌ی اضافه‌ی بعد از «ه» غیرملفوظ به صورت «ه‌ی» یا «هٔ» نوشته می‌شود. مانند: خانه‌ی علی، دنباله‌ی فیبوناچی.

تبصره: اگر «ه» ملفوظ باشد، نیاز به «ی» ندارد. مانند: فرمانده دلیر، پادشه خوبان.

۴. پایه‌های همزه در کلمات، همیشه «ئ» است، مانند: مسئله و مسئول، مگر در مواردی که همزه ساکن است که در این صورت باید متناسب با اعراب حرف پیش از خود نوشته شود. مانند: رأس، مؤمن.

۲ـ۲ـ۳ جدانویسی

۱. علامت استمرار، «می»، توسط نیم‌فاصله از جزء بعدی فعل جدا می‌شود. مانند: می‌رود، می‌توانیم.

۲. شناسه‌های «ام»، «ای»، «ایم»، «اید» و «اند» توسط نیم‌فاصله، و شناسه‌ی «است» توسط فاصله از کلمه‌ی پیش از خود جدا می‌شوند. مانند: گفته‌ام، گفته‌ای، گفته است.

۳. علامت جمع «ها» توسط نیم‌فاصله از کلمه‌ی پیش از خود جدا می‌شود. مانند: این‌ها، کتاب‌ها.

۴. «به» همیشه جدا از کلمه‌ی بعد از خود نوشته می‌شود، مانند: به نام و به آن‌ها، مگر در مواردی که «بـ» صفت یا فعل ساخته است. مانند: بسزا، ببینم.

۵. «به» همواره با فاصله از کلمه‌ی بعد از خود نوشته می‌شود، مگر در مواردی که «به» جزئی از یک اسم یا صفت مرکب است. مانند: تناظر یک‌به‌یک، سفر به تاریخ.

۶. علامت صفت برتری، «تر»، و علامت صفت برترین، «ترین»، توسط نیم‌فاصله از کلمه‌ی پیش از خود جدا می‌شوند. مانند: سنگین‌تر، مهم‌ترین.

تبصره: کلمات «بهتر» و «بهترین» را می‌توان از این قاعده مستثنی نمود.

۷. پیشوندها و پسوندهای جامد، چسبیده به کلمه‌ی پیش یا پس از خود نوشته می‌شوند. مانند: همسر، دانشکده، دانشگاه.

تبصره: در مواردی که خواندن کلمه دچار اشکال می‌شود، می‌توان پسوند یا پیشوند را جدا کرد. مانند: هم‌میهن، هم‌ارزی.

[۳] صفحه‌کلید فارسی استاندارد برای ویندوز، تهیه‌شده توسط بهنام اسفهبد

۸. ضمیرهای متصل چسبیده به کلمه‌ی پیش از خود نوشته می‌شوند. مانند: کتابم، نامت، کلامشان.

فصل ۳

کارهای پیشین

در فصل سوم پایان‌نامه، کارهای پیشین انجام‌شده روی مسئله به تفصیل توضیح داده می‌شود. نمونه‌ای از فصل کارهای پیشین در زیر آمده است.[۱]

۳ـ۱ مسائل خوشه‌بندی

مسئله‌ی **خوشه‌بندی**[۲] یکی از مهم‌ترین مسائل در زمینه‌ی داده‌کاوی به حساب می‌آید. در این مسئله، هدف دسته‌بندی تعدادی شیء به‌گونه‌ای است که اشیاء درون یک دسته (خوشه)، نسبت به یکدیگر در برابر دسته‌های دیگر شبیه‌تر باشند (معیارهای متفاوتی برای تشابه تعریف می‌گردد). این مسئله در حوزه‌های مختلفی از علوم کامپیوتر از جمله داده‌کاوی، جست‌وجوی الگو[۳]، پردازش تصویر[۴]، بازیابی اطلاعات[۵] و رایانش زیستی[۶] مورد استفاده قرار می‌گیرد [؟].

تا کنون راه‌حل‌های زیادی برای این مسئله ارائه شده است که از لحاظ معیار تشخیص خوشه‌ها و نحوه‌ی انتخاب یک خوشه، با یک‌دیگر تفاوت بسیاری دارند. به همین خاطر مسئله‌ی خوشه‌بندی یک مسئله‌ی بهینه‌سازی چندهدفه[۷] محسوب می‌شود.

همان طور که در مرجع [؟] ذکر شده است، خوشه در خوشه‌بندی تعریف واحدی ندارد و یکی از

---

[۱] مطالب این فصل نمونه از پایان‌نامه‌ی آقای بهنام حاتمی گرفته شده است.
[۲] Clustering
[۳] Pattern recognition
[۴] Image analysis
[۵] Information retrieval
[۶] Bioinformatics
[۷] Multi-objective

دلایل وجود الگوریتم‌های متفاوت، همین تفاوت تعریف‌ها از خوشه است. بنابراین با توجه به مدلی که برای خوشه‌ها ارائه می‌شود، الگوریتم متفاوتی نیز ارائه می‌گردد. در ادامه به بررسی تعدادی از معروف‌ترین مدل‌های مطرح می‌پردازیم:

- **مدل‌های مرکزگرا**: در این مدل‌ها، هر دسته با یک مرکز نشان داده می‌شود. از جمله معروف‌ترین روش‌های خوشه‌بندی بر اساس این مدل، خوشه‌بندی kـ مرکز، خوشه‌بندی kـ میانگین[۸] و خوشه‌بندی kـ میانه[۹] است.

- **مدل‌های مبتی بر توزیع نقاط**: در این مدل، دسته‌ها با فرض پیروی از یک توزیع احتمالی مشخص می‌شوند. از جمله الگوریتم‌های معروف ارائه شده در این مدل، الگوریتم بیشینه‌سازی امید ریاضی[۱۰] است.

- **مدل‌های مبتنی بر تراکم نقاط**: در این مدل، خوشه‌ها متناسب با ناحیه‌های متراکم نقاط در مجموعه داده مورد استفاده قرار می‌گیرد.

- **مدل‌های مبتنی بر گراف**: در این مدل، هر خوشه به مجموعه از رئوس گفته می‌شود که تمام رئوس آن با یک‌دیگر همسایه باشند. از جمله الگوریتم‌های معروف این مدل، الگوریتم خوشه‌بندی HCS[۱۱] است.

الگوریتم‌های ارائه شده تنها از نظر نوع مدل با یک‌دیگر متفاوت نیستند. بلکه، می‌توان آن‌ها را از لحاظ نحوه‌ی تخصیص نقاط بین خوشه‌ها نیز تقسیم‌بندی کرد:

- **تخصیص قطعی داده‌ها**: در این نوع خوشه‌بندی هر داده دقیقاً به یک خوشه اختصاص داده می‌شود.

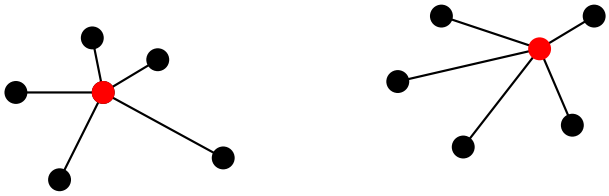
- **تخصیص قطعی داده‌ها با داده‌ی پرت**: در این نوع خوشه‌بندی ممکن است بعضی از داده‌ها به هیچ خوشه‌ای اختصاص نیابد، اما بقیه داده‌ها هر کدام دقیقاً به یک خوشه اختصاص می‌یابد.

- **تخصیص قطعی داده**: در این نوع خوشه‌بندی هر داده دقیقاً به یک خوشه اختصاص داده می‌شود.

- **خوشه‌بندی هم‌پوشان**: در این نوع خوشه‌بندی هر داده می‌تواند به چند خوشه اختصاص داده شود. در گونه‌ای از این مدل، می‌توان هر نقطه را با احتمالی به هر خوشه اختصاص می‌یابد. به این گونه از خوشه‌بندی، خوشه‌بندی نرم[۱۲] گفته می‌شود.

---

[۸] k-Means
[۹] k-Median
[۱۰] Expectation-maximization
[۱۱] Highly Connected Subgraphs
[۱۲] Soft clustering



شکل ۳ـ۱: نمونه‌ای از مسئله‌ی ۲ـ مرکز

- **خوشه‌بندی سلسه‌مراتبی**: در این نوع خوشه‌ها، داده‌ها به گونه‌ای به خوشه‌ها تخصیص داده می‌شود که دو خوشه یا اشتراک ندارند یا یکی به طور کامل دیگری را می‌پوشاند. در واقع در بین خوشه‌ها، رابطه‌ی پدر فرزندی برقرار است.

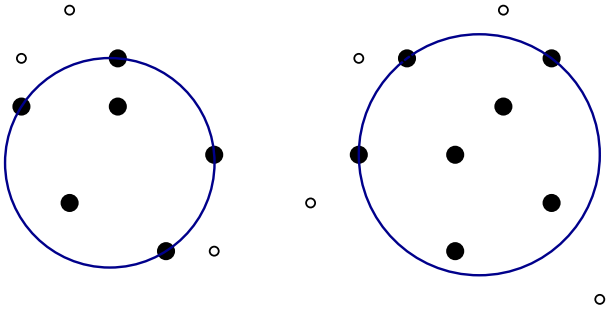
در بین دسته‌بندی‌های ذکر شده، تمرکز اصلی این پایان‌نامه بر روی مدل مرکزگرا و خوشه‌بندی قطعی با داده‌های پرت با مدل kـ مرکز است. همان‌طور که ذکر شد علاوه بر مسئله‌ی kـ مرکز که به تفصیل مورد بررسی قرار می‌گیرد، kـ میانه و kـ میانگین از جمله معروف‌ترین خوشه‌بندی‌های مدل مرکزگرا هستند. در خوشه‌بندی kـ میانه، هدف افراز نقاط به k خوشه است به گونه‌ای که مجموع مربع فاصله‌ی هر نقطه از میانه‌ی نقاط آن خوشه، کمینه گردد. در خوشه‌بندی kـ میانگین، هدف افراز نقاط به k خوشه است به گونه‌ای که مجموع فاصله‌ی هر نقطه از میانگین نقاط داخل خوشه (یا مرکز آن خوشه) کمینه گردد.

۳ـ۲ خوشه‌بندی kـ مرکز

یکی از رویکردهای شناخته‌شده برای مسئله‌ی خوشه‌بندی، مسئله‌ی ***k*ـ مرکز** است. در این مسئله هدف، پیدا کردن k نقطه به عنوان مرکز دسته‌ها است به‌طوری‌که شعاع دسته‌ها تا حد ممکن کمینه شود. مثالی از مسئله‌ی ۲ـ مرکز در شکل ۳ـ۱ نشان داده شده است. در این پژوهش، مسئله‌ی kـ مرکز با متریک‌های خاص و برای kهای کوچک مورد بررسی قرار گرفته است و هر کدام از تعریف رسمی مسئله‌ی kـ مرکز در زیر آمده است:

مسئله‌ی ۳ـ۱ (kـ **مرکز**) *گراف کامل بدون جهت* $G = (V, E)$ *با تابع فاصله‌ی* d*، که از نامساوی مثلثی پیروی می‌کند داده شده است. زیرمجموعه‌ی* $S \subseteq V$ *با اندازه‌ی* k *را به‌گونه‌ای انتخاب کنید که عبارت زیر را کمینه کند:*

$$\max_{v \in V} \{\min_{s \in S} d(v, s)\} \qquad (۱ـ۳)$$



شکل ۳ـ۲: نمونه‌ای ازمسئله‌ی ۲ـ مرکز با داده‌های پرت

گونه‌های مختلفی از مسئله‌ی kـ مرکز با محدودیت‌های متفاوت توسط پژوهشگران مورد مطالعه قرار گرفته است. از جمله‌ی این گونه‌ها، می‌توان به حالتی که در بین داده‌های ورودی، داده‌های پرت وجود دارد، اشاره کرد. در واقع در این مسئله، قبل از خوشه‌بندی می‌توانیم تعدادی از نقاط ورودی را حذف نموده و سپس به خوشه‌بندی نقاط بپردازیم. سختی این مسئله از آنجاست که نه تنها باید مسئله‌ی خوشه‌بندی را حل نمود، بلکه در ابتدا باید تصمیم گرفت که کدام یک از داده‌ها را به‌عنوان داده‌ی پرت در نظر گرفت که بهترین جواب در زمان خوشه‌بندی به دست آید. در واقع اگر تعداد نقاط پرتی که مجاز به حذف است، برابر صفر باشد، مسئله به مسئله‌ی kـ مرکز تبدیل می‌شود. نمونه‌ای از مسئله‌ی ۲ـ مرکز با ۷ داده‌ی پرت را در شکل ۳ـ۲ می‌توانید ببینید. تعریف دقیق‌تر این مسئله در زیر آمده است:

مسئله‌ی ۳ـ۲ (**kـ مرکز با داده‌های پرت**) *یک گراف کامل بدون جهت $G = (V, E)$ با تابع فاصله‌ی d، که از نامساوی مثلثی پیروی می‌کند داده‌شده است. زیرمجموعه‌ی $Z \subseteq V$ با اندازه‌ی z و مجموعه‌ی $S \subseteq V - Z$ با اندازه‌ی k را انتخاب کنید به طوری‌که عبارت زیر را کمینه کند:*

$$\max_{v \in V-Z} \{\min_{s \in S} d(v, s)\} \qquad (۲ـ۳)$$

گونه‌ی دیگری از مسئله‌ی kـ مرکز که در سال‌های اخیر مورد توجه قرار گرفته است، حالت جویبار داده‌ی آن است. در این‌گونه از مسئله‌ی kـ مرکز، در ابتدا تمام نقاط در دسترس نیستند، بلکه به‌مرور زمان نقاط در دسترس قرار می‌گیرند. محدودیت دومی که وجود دارد، محدودیت حافظه است، به‌طوری‌که نمی‌توان تمام نقاط را در حافظه نگه داشت و بعضاً حتی امکان نگه‌داری در حافظه‌ی جانبی نیز وجود ندارد و به‌طور معمول باید مرتبه‌ی حافظه‌ای کم‌تر از مرتبه حافظه‌ی **خطی**[۱۳] متناسب با تعداد نقاط استفاده نمود. از این به بعد به چنین مرتبه‌ای، مرتبه‌ی **زیرخطی**[۱۴] می‌گوییم. مدلی که ما در این پژوهش بر روی آن تمرکز داریم مدل جویبار داده تک‌گذره[۱۵] [؟] است. یعنی تنها یک بار می‌توان از ابتدا تا انتهای داده‌ها را بررسی

[۱۳] Linear
[۱۴] sublinear
[۱۵] Single pass

کرد و پس از عبور از یک داده، اگر آن داده در حافظه ذخیره نشده باشد، دیگر به آن دسترسی وجود ندارد. علاوه بر این، در هر لحظه باید بتوان به پرسمان (برای تمام نقاطی از جویبار داده که تاکنون به آن دسترسی داشته‌ایم) پاسخ داد.

مسئله‌ی ۳ـ۳ (kـ**مرکز در حالت جویبار داده**) *مجموعه‌ای از نقاط در فضای dـ بعدی به مرور زمان داده می‌شود. در هر لحظه از زمان، به ازای مجموعه‌ی U از نقاطی که تا کنون وارد شده‌اند، زیرمجموعه‌ی $S \subseteq U$ با اندازه‌ی k را انتخاب کنید به طوری‌که عبارت زیر کمینه شود:*

$$\max_{u \in U} \{\min_{s \in S} d(u, s)\} \qquad (۳ـ۳)$$

از آنجایی که گونه‌ی جویبار داده و داده پرت مسئله‌ی kـ مرکز به علت به‌روز بودن مبحث داده‌های حجیم[۱۶]، به تازگی مورد توجه قرار گرفته است. در این تحقیق سعی شده است که تمرکز بر روی این‌گونه‌ی خاص از مسئله باشد. همچنین در این پژوهش سعی می‌شود گونه‌های مسئله را برای انواع متریک‌ها و برای kهای کوچک نیز مورد بررسی قرار داد.

۳ـ۳ مدل جویبار داده

همان‌طور که ذکر شد مسئله‌ی kـ مرکز در حالت داده‌های پرت و جویبار داده، گونه‌های تعمیم‌یافته از مسئله‌ی kـ مرکز هستند و در حالت‌های خاص به مسئله‌ی kـ مرکز کاهش پیدا می‌کنند. مسئله‌ی kـ مرکز در حوزه‌ی مسائل ان‌پی‌ـ سخت[۱۷] قرار می‌گیرد و با فرض $P \neq NP$ الگوریتم دقیق با زمان چندجمله‌ای برای آن وجود ندارد [؟]. بنابراین برای حل کارای[۱۸] این مسائل از الگوریتم‌های تقریبی[۱۹] استفاده می‌شود.

برای مسئله‌ی kـ مرکز، دو الگوریتم تقریبی معروف وجود دارد. در الگوریتم اول، که به روش حریصانه[۲۰] عمل می‌کند، در هر مرحله بهترین مرکز ممکن را انتخاب می‌کند به طوری تا حد ممکن از مراکز قبلی دور باشد [؟]. این الگوریتم، الگوریتم تقریبی با ضریب تقریب ۲ ارائه می‌دهد. در الگوریتم دوم، با استفاده از مسئله‌ی مجموعه‌ی غالب کمینه[۲۱]، الگوریتمی با ضریب تقریب ۲ ارائه می‌گردد [؟]. همچنین ثابت شده است، که بهتر از این ضریب تقریب، الگوریتمی نمی‌توان ارائه داد مگر آن‌که $P = NP$ باشد.

---
[۱۶] Big data
[۱۷] NP-hard
[۱۸] Efficient
[۱۹] Approximation algorithm
[۲۰] Greedy
[۲۱] Dominating set

جدول ۳ـ۱: نمونه‌هایی از کران پایین تقریب‌پذیری مسائل خوشه‌بندی

مسئله	کران پایین تقریب‌پذیری
kـ مرکز	۲ [؟]
kـ مرکز در فضای اقلیدسی	۱/۸۲۲ [؟]
۱ـ مرکز در حالت جویبار داده	$\frac{۱+\sqrt{۲}}{۲}$ [؟]
kـ مرکز با نقاط پرت و نقاط اجباری	۳ [؟]

برای مسئله‌ی kـ مرکز در حالت جویبار داده برای ابعاد بالا، بهترین الگوریتم موجود ضریب تقریب $۲+\epsilon$ دارد [؟، ؟، ؟] و ثابت می‌شود الگوریتمی با ضریب تقریب بهتر از ۲ نمی‌توان ارائه داد. برای مسئله‌ی kـ مرکز با داده‌ی پرت در حالت جویبار داده نیز، بهترین الگوریتم ارائه شده، الگوریتمی با ضریب تقریب $۴+\epsilon$ است که با کران پایین ۳ هنوز اختلاف قابل توجهی دارد [؟].

برای kهای کوچک به خصوص، $k = ۱, ۲$، الگوریتم‌های بهتری ارائه شده است. بهترین الگوریتم ارائه شده برای مسئله‌ی ۱ ـ مرکز در حالت جویبار داده برای ابعاد بالا، دارای ضریب تقریب ۱/۲۲ است و کران پایین $\frac{۱+\sqrt{۲}}{۲}$ نیز برای این مسئله اثبات شده است [؟، ؟]. برای مسئله ۲ ـ مرکز در حالت جویبار داده برای ابعاد بالا، اخیرا راه‌حلی با ضریب تقریب $۱/۸+\epsilon$ ارائه شده است [؟]. برای مسئله‌ی ۱ ـ مرکز با داده‌ی پرت، تنها الگوریتم موجود، الگوریتمی با ضریب تقریب ۱/۷۳ است [؟].

۳ـ۴ تقریب‌پذیری

یکی از راه‌کارهایی که برای کارآمد کردن راه‌حل ارائه شده برای یک مسئله وجود دارد، استفاده از الگوریتم‌های تقریبی برای حل آن مسئله است. یکی از عمده‌ترین دغدغه‌های مطرح در الگوریتم‌های تقریبی کاهش ضریب تقریب است. در بعضی از موارد حتی امکان ارائه‌ی الگوریتم تقریبی با ضریبی ثابت نیز وجود ندارد. به طور مثال، الگوریتم تقریبی با ضریب تقریب کم‌تر از ۲، برای مسئله‌ی kـ مرکز وجود ندارد مگر این‌که $P = NP$ باشد. برای مسائل مختلف، معمولاً می‌توان کران پایینی برای میزان تقریب‌پذیری آن‌ها ارائه داد. در واقع برای برخی مسائل انپی‌ـ سخت، علاوه بر این که الگوریتم کارآمدی وجود ندارد، بعضاً الگوریتم تقریبی با ضریبی تقریب کم و نزدیک به یک نیز وجود ندارد. در جدول ۳ـ۱ میزان تقریب‌پذیری مسائل مختلفی که در این پایان‌نامه مورد استفاده قرار می‌گیرد را می‌بینید.

فصل ۴

نتایج جدید

در این فصل نتایج جدید به‌دست‌آمده در پایان‌نامه توضیح داده می‌شود. در صورت نیاز می‌توان نتایج جدید را در قالب چند فصل ارائه نمود. همچنین در صورت وجود پیاده‌سازی، بهتر است نتایج پیاده‌سازی را در فصل مستقلی پس از این فصل قرار داد.

فصل ۵

نتیجه‌گیری

در این فصل، ضمن جمع‌بندی نتایج جدید ارائه‌شده در پایان‌نامه یا رساله، مسائل باز باقی‌مانده و همچنین پیشنهادهایی برای ادامه‌ی کار ارائه می‌شوند.

فصل ۶

نتیجه‌گیری

مراجع

[1] Global mobile OS market share 2022 | Statista — statista.com. https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/#:~:text=Android%20maintained%20its%20position%20as,the%20mobile%20operating%20system%20market. [Accessed 02-Feb-2023].

[2] Play Protect | Google Developers — developers.google.com. https://developers.google.com/android/play-protect. [Accessed 02-Feb-2023].

[3] Decompile and modify an Android application | cylab.be — cylab.be. https://cylab.be/blog/69/decompile-and-modify-an-android-application. [Accessed 02-Feb-2023].

[4] A. Dizdar. OWASP Mobile Top 10 Vulnerabilities and How to Prevent Them — brightsec.com. https://brightsec.com/blog/owasp-mobile-top-10/. [Accessed 02-Feb-2023].

[5] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee, and K. P. Wu. DroidMat: Android malware detection through manifest and API calls tracing. *Proceedings of the 2012 7th Asia Joint Conference on Information Security, AsiaJCIS 2012*, pages 62–69, 2012.

[6] K. Khanmohammadi, N. Ebrahimi, A. Hamou-Lhadj, and R. Khoury. Empirical study of android repackaged applications. *Empirical Software Engineering*, 24(6):3587–3629, 2019.

[7] T. Vidas and N. Christin. Sweetening android lemon markets: Measuring and combating malware in application marketplaces. *CODASPY 2013 - Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy*, 2011:197–207, 2013.

[8] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury. A study on malicious software behaviour analysis and detection techniques: Taxonomy, current trends and challenges. *Future Generation Computer Systems*, 130:1–18, 2022.

[9] Z. Ma, H. Wang, Y. Guo, and X. Chen. Libradar: Fast and accurate detection of third-party libraries in android apps. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pages 653–656, 2016.

واژه‌نامه

الف

ابتکاری ..... heuristic
ابعاد بالا ..... high dimensions
اریب ..... bias
آستانه ..... threshold
اصل لانه‌ی کبوتری ..... pigeonhole principle
ان‌پی‌ـسخت ..... NP-Hard
انتقال ..... transition

ب

برخط ..... online
برنامه‌ریزی خطی ..... linear programming
بهینه ..... optimum
بیشینه ..... maximum

پ

پرت ..... outlier
پرسمان ..... query
پوشش ..... cover
پیچیدگی ..... complexity

ت

تجربی ..... experimental
تراکم ..... density
تقریب ..... approximation
تقسیم‌بندی ..... partition
توری ..... mesh
توزیع‌شده ..... distributed

ج

جداپذیر ..... separable
جعبه سیاه ..... black box
جویبار داده ..... data stream

ح

حدی ..... extreme
حریصانه ..... greedy

خ

خوشه ..... cluster
خطی ..... linear

د

داده ..... data

داده‌کاوی ..... data mining

داده‌ی پرت ..... outlier data

دوبرابرسازی ..... doubling

دودویی ..... binary

ر

رأس ..... vertex

رسمی ..... formal

ز

زیرخطی ..... sublinear

س

سرشکن ..... amortized

سلسه‌مراتبی ..... hierarchichal

ش

شبه کد ..... pseudocode

شیء ..... object

ص

صدق‌پذیری ..... satisfiability

غ

غلبه ..... dominate

ف

فاصله ..... distance

فضا ..... space

ق

قطعی ..... deterministic

ک

کارا ..... efficient

کاندیدا ..... candidate

کمینه ..... minimum

م

مجموعه ..... set

مجموعه هسته ..... coreset

مسطح ..... planar

موازی‌سازی ..... parallelization

میان‌گیر ..... buffer

ن

نابه‌جایی ..... inversion

ناوردا ..... invariant

نقطه‌ی مرکزی ..... center point

نیم‌فضا ..... half space

هـ

هزینه‌ی آشوب ..... price of anarchy (POA)

ی

یال ..... edge

پیوست آ

مطالب تکمیلی

پیوست‌های خود را در صورت وجود می‌توانید در این قسمت قرار دهید.

Abstract

We present a standard template for typesetting theses in Persian. The template is based on the XePersian package for the LaTeX typesetting system. This write-up shows a sample usage of this template.

Keywords: Thesis, Typesetting, Template, XePersian

Sharif University of Technology

Department of Computer Engineering

M.Sc. Thesis

Performance Improvement of Android Repackaged Applications

By:

Mojtaba Moazen

Supervisor:

Dr. Amini

february 2023