پایان‌نامه دوره کارشناسی

مهندسی کامپیوتر- گرایش نرم افزار

عنوان پروژه:

طراحی و پیاده‌سازی سیستم پیشنهاد دهنده اشیا در اینترنت اشیا

دانشجو:

زهرا ولدی

استاد راهنما:

دکتر کیانیان

بهمن ۱۴۰۱


بسم الله الرحمن الرحيم

تأییدیه اتمام پروژه

گواهی می‌شود که زهرا ولدی با شماره دانشجویی ۳۹۷۱۲۳۱۰۹۸ دانشجوی رشته مهندسی کامپیوتر گرایش نرم‌افزار دانشکده مهندسی کامپیوتر پایان‌نامه مقطع کارشناسی که دارای ۳ واحد بوده است را با عنوان طراحی و پیاده‌سازی سیستم پیشنهاد دهنده اشیا در اینترنت اشیا تحت نظارت استاد سحر کیانیان با نمره ..... در تاریخ ..... به اتمام رسانده‌اند. نسخه نهایی پایان‌نامه و فایل آن (به‌صورت DOCX و PDF) مطابق با ساختار کلی و دستورالعمل مصوب دانشکده تهیه و تحویل آموزش دانشکده شده است

.

ردیف	عنوان	نام و نام خانوادگی	امضا
۱	استاد راهنمای پروژه		
۲	مسئول پروژه‌ها گروه		
۳	مدیر گروه		
۴	رئیس یا معاون آموزشی و پژوهشی دانشکده		

چکیده

تحول ایجاد شده در دنیای تکنولوژی باعث به وجود آمدن مفهوم جدیدی به نام اینترنت اشیا شده است که در آن تعدادی اشیاء تحت شرایطی خاص نیازهای کاربران را بصورت آنلاین برآورده می‌نمایند. هرکدام از این اشیا تحت کنترل تعدادی سرویس دهنده می‌باشند و به اصطلاح این سرویس دهنده‌ها هر کدام به تعدادی از اشیا سرویس می‌دهند. کاربران با توجه به نوع نیاز شان هر کدام تعدادی از خدمات ارائه شده توسط این سرویس دهنده‌ها را بکار میگیرند. در این میان مسئله‌ای که قابل بحث است در مورد پیشنهاد سرویس دهنده‌هایی است که برای کاربران استفاده از آنها بهینه‌تر و مفیدتر باشد. در همین راستا در این تحقیق سیستمی توصیه گر طراحی شده که بر اساس روابط بین کاربران، انواع اشیا و سرویس دهنده‌ها به کاربران پیشنهاد استفاده از سرویس دهنده‌ای را می‌دهد که در راستای نیازمندی‌های کاربران باشد و از این طریق بتواند به کاربران پیشنهاد استفاده از سرویس دهنده مورد نظر را ارائه نماید. در این سیستم توصیه گر پیشنهادی ویژگی های هر کدام از کاربران و سرویس دهنده های موجود را بررسی کرده و به تشخیص ارتباط و نیازهای کاربران پرداخته و سرویس دهنده‌ای به کاربر پیشنهاد داده می‌شود که بر اساس ویژگی‌های کاربران و در راستای برآورده شدن این نیازها باشد.

کلمات کلیدی

اینترنت اشیا ، سیستم های توصیه گر ، سیستم پیش بینی کننده

فهرست مطالب

فهرست جدول‌ها

فهرست شکل‌ها

فصل ۱

مقدمه

۱-۱- مقدمه

کند و کاو در اینترنت ا شیا[۱] برای د ستیابی به خدمات و اطلاعات به رو شی قابل اعتماد برای مدتی طولانی یک چالش بوده ا ست. راه حل های مختلفی برای مقابله با این چالش ارائه شده ا ست. با این حال، با توجه به افزایش تعداد دستگاه های اینترنت اشیا با نرخی بالا این راه حل ها چندان کارآمد نیستند.ادغام ویژگی های شبکه های اجتماعی با الگوهای اینترنت اشیا به منظور غلبه بر مشکلات موجود در اینترنت اشیا توجهات بی سابقه ای را به خود جلب کرده است. تلاش های زیادی برای ادغام دستگاه های اینترنت اشیا در حلقه های اجتماعی[۲] انجام شده است مانند پروژه های وبلاگ[۳][2]، روش دوست هوشمند[۴] [3] و پروژه اریکسون[۵][1,4].

[۱] Iot

[۲] Social loops

[۳] Blog-jects

[۴] Smart-Its friend procedure

[۵] Ericson project

مفهومی جدید به نام اینترنت اجتماعی ا شیا[1] از این موارد پدید آمده ا ست و هدف ا صلی آن اجازه دادن به اشیا برای ایجاد ارتباط بایکدیگر با توجه به ترجیهات صاحبان این دستگاه ها[1, 5, 6, 7].

چ شم انداز اینترنت اجتماعی ا شیا گنجاندن رفتار های اجتماعی در د ستگاه های هو شمند اینترنت ا شیا و اجازه دادن به اشیا برای داشتن شبکه اجتماعی خودگران.

مزایای زیادی در اجتماعی ساختن اینترنت اشیا وجود دارد. نخست اینکه اینترنت اجتماعی اشیا می‌تواند در د سترس بودن منابع را تقویت کند و با استفاده از دوستان و دوستان دوستان کشف خدمات[2] را به گونه ی توزیع شده[3] افزایش دهد[8] ، برخلاف اینترنت اشیا سنتی که خدمات به گونه ایی متمرکز[4] توسط موتورهای جستجو کشف می‌شدند. دوم اینکه، شیوه ی متمرکز یافتن اشیا مشکلات مقیاس پذیری را مطرح می‌کند که در اینترنت اجتماعی ا شیا چون هر د ستگاه اینترنت ا شیا ساختار شبکه اینترنت اجتماعی ا شیا را برای د ست‌یابی به د ستگاه های دیگر هدایت می‌کند، این م شکل نیز حل می شود[5, 9, 10]. سوم اینکه، براساس ساختار اجتماعی ایجاد شده میان دستگاه های اینترنت اشیا،دستگاه ها می‌توانند برای ارزیابی اعتبار دستگاه های دیگر یک همسایگی[5] محلی ایجاد کنند. چهارم اینکه ، با استفاده از اینترنت اجتماعی ا شیا د ستگاه ها می‌توانند نسبت به د ستگاه های دیگر آگاهی پیدا کرده که بتوانند اطلاعات و تجربیات را با همدیگر مبادله کنند[8,1].

فعالیت های زیادی برای تحقق بخ شیدن به مفهوم اینترنت اجتماعی آ شیا انجام شده ا ست. با این حال اکثر این فعالیت‌های پژوه شی بر شنا سایی سیا ست‌ها،روش ها و راه کارها برای برقراری ارتباط بین

[1] SIot

[2] Service discovery

[3] Distributed manner

[4] Centralized

[5] Neighborhood

دستگاه های هوشمند به طور مستقل و بدون دخالت انسان متمرکز بوده است[5, 8]. علاوه بر این معماری های مختلفی از اینترنت اجتماعی اشـیا ارائه شـده اسـت[5, 11, 12]. علی رغم تحقیقات فشـرده بر روی اینترنت اجتماعی اشیا ملاحظات کافی در مدل سازی و تحلیل شبکه اینترنت اجتماعی اشیا ارائه شده وجود ندارد. طبیعت اینترنت اجتماعی اشیا به شکل پویا است به این صورت که با گذشت زمان رشد و تغییر میکند رئوس گراف[1](دستگاه‌های اینترنت اشیا) و یال‌های گراف(روابط بین دستگاه‌ها) به طور مداوم پدید می‌آیند و ناپدید می شوند. بنابراین علاقه مندی زیادی برای توسعه مدل‌هایی که امکان مطالعه ودرک این شبکه های در حال تحول و به ویژه پیش بینی ایجاد پیوند های آینده[2] به وجود آمده اســـت[6, 8]. پیش بینی پیوند های آینده بین اشیا می‌تواند برای سیستم های توصیه گر و کشف سرویس‌ها استفاده شود.بنابراین نیاز به شناسایی ساز و کاری که اینترنت اجتماعی اشیا بتواند با استفاده از آن تکامل پیدا کند وجود دارد.این مسئله یک چالش بنیادین در اینترنت اجتماعی اشیا است که هنوز به اندازه کافی به آن پرداخته نشده است و هدف کار انجام شده در این گزارش پاسخ‌گویی به این چالش است[1].

[1] Graph nodes

[2] Link prediction

فصل ۲

پیشینه

۲-۱- مقدمه

اینترنت اشیا شبکه ای از دستگاه ها، اشیا و ماشین های ناهمگون که هر کدام به طور منحصر به فرد قابل شناسایی و به هم متصلند است، که بدون نیاز به ارتباط انسان با رایانه یا انسان با انسان می‌توانند داده ها را ارائه دهند. براساس تحقیقات پیشین [14] نشان داده می‌دهد که تعداد دستگاه های متصل به هم در سال ۲۰۱۹، ۱۴ میلیارد بوده و انتظار می‌رود تعداد این دستگاه ها به ۲۵ میلیارد برسد[13] .

۲-۲- اینترنت اجتماعی اشیا

با الهام از نظریه فیسک[۱] [15] که براساس روابط اجتماعی میان انسان هست، ساختار اجتماعی سازی اینترنت اشیا معرفی می شود. فیسک با مطالعه طبیعت روابط میان انسان ها یک مدل برای ارتباطات اجتماعی ارائه داد. این مدل قابل استفاده برای ارتباط میان اشیا به صورت اشتراک گذاری منابع،طبقه بندی اختیار میان اشیا و همکاری سودمند متقابل بین دستگاه ها باشد[13].

در محیط اینترنت اجتماعی اشیا هر دستگاه می‌تواند هم به عنوان فراهم آورنده و هم درخواست کننده سرویس و اطلاعات عمل کند[16]. تعداد بسیار زیاد خدمات مبادله شده بین دستگاه های مختلف چالشی را برای انتخاب سرویس های مناسب ایجاد می‌کند که به همین دلیل نیاز سیستم های پیشنهاد دهنده احساس می‌شود. در این رویکرد ما از روابط اجتماعی تعریف شده در اینترنت اجتماعی اشیا برای ایجاد پیشنهاد دهنده ی سرویس ها بین اشیا و افزایش کشف سرویس ها و ترکیب آن ها استفاده می‌کنیم[13].

[۱] Fiske's theory

۲-۲-۱-روابط اجتماعی میان دستگاه‌ها

الگوی محاسباتی اینترنت اجتماعی اشیا ناوبری شبکه[۱] و ترکیب خدمات[۲] را با متعامل کردن اجتماعی دستگاه ها بهبود می‌بخشد. روابط متفاوت ذکر شده در اینترنت اشیا سناریو[۳] هایی را ایجاد می‌کند که می‌توان از تعامل اجتماعی اشیا برای آن‌ها استفاده کرد. به منظور بهبود انتخاب سرویس[۴] و ترکیب سرویس ، دستگاه‌ها می‌توانند با درخواست کنندگان و فراهم آورندگان خدمات تعامل و تطابق داشته باشند.علاوه براین دستگاه‌ها می‌توانند برای فراهم آوردن سفارشی سازی، پیکر بندی و تبادل خدمات مربوطه بین یکدیگر براساس اعتماد،دقت و درستی با یکدیگر همکاری کنند.چند نوع از روابط میان دستگاه‌های اینترنت اشیا در جدول ۲-۱ ذکر شده است[13].

جدول ۰-۱: تعدادی از روابط اجتماعی میان اشیا

پویایی یا ایستایی رابطه	توضیحات	نوع رابطه میان دستگاه های اینترنت اجتماعی اشیا
ایستا	این رابطه میان دستگاه‌های هم نسل تولید شده توسط یک تولید کننده به وجود می‌آید	دستگاه و والد[۵]
پویا	این رابطه میان دستگاه‌هایی که همیشه در یک مکان هستند ایجاد می‌شود	دستگاه های هم مکان[۶]

[۱] Network navigability

[۲] Service composition

[۳] Scenario

[۴] Service selection

[۵] Parental object relationship(POR)

[۶] Co-location object relationship(CLOR)

پویا	این رابطه میان دستگاه‌هایی که برای دستیابی به یک هدف خاص یا ارائه یک برنامه کاربردی خاص همکاری می‌کنند ایجاد می‌شود	دستگاه های همکار[1]
ایستا	این رابطه میان دستگاه‌هایی که صاحب آن‌ها یک نفر است ایجاد می‌شود	مالکیت دستگاه[2]
پویا	این رابطه میان دستگاه‌هایی که به طور مداوم یا پراکنده (خودشان یا صاحبانشان در طول کارهای روزمره با هم در تماس هستند)با هم در ارتباط هستند ایجاد می‌شود	دستگاه های اجتماعی[3]

۲-۳ –پیش بینی پیوندهای آینده

هدف پیش‌بینی پیوندهای آینده تخمین احتمال ایجاد هر کدام از یال‌های غیرمتصل در یک شبکه به منظور شناسایی مجموعه‌ای از پیوندهای آینده یا گم شده بین کاربران است.بسیاری از وظایف مهم در تجزیه و تحلیل شبکه ها با استفاده از پیش بینی رئوس و یال ها در شبکه‌ها امکان می‌پذیرد.در یک وظیفه ساده طبقه بندی گره‌ها، ما علاقه مند به پیش بینی محتمل ترین برچسب‌های احتمالی[4] یک گره در شبکه هستیم[18]. برای مثال در شبکه اینترنت اجتماعی اشیا ما به دنبال یافتن محتمل ترین گره(دستگاه)کاربردی هستیم . به طور مشابه در پیش‌بینی پیوندهای آینده تلاش می‌کنیم که بدانیم امکان ایجاد شدن یالی بین یک جفت گره در شبکه وجود دارد.پیش بینی لینک‌های آینده در زمینه‌های مختلف به طور مثال: ژنومیکس[5] که برای یافتن تعاملات جدید ژن‌ها از پیش پیوندهای آینده می‌توان استفاده کرد یا در شبکه اینترنت اجتماعی اشیا برای یافتن خدمات جدید استفاده می‌شود[17].

[1] Co-work object relationship(CWOR)

[2] Ownership object relationship(OOR)

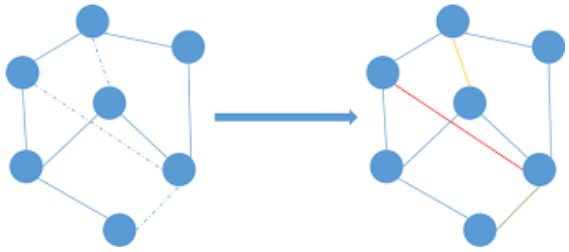
[3] Social object relationship(SOR)

[4] Probable labels

[5] genomics

هر الگوریتم یادگیری ماشـین[۱] نیازمند مجموعه ای از اطلاعات اولیه و ویژگی‌های منحصـر به فرد اسـت.در مشـکلات موجود در پیش‌بینی پیوندهای آینده این موضـوع به این معنا اسـت که باید یک جدول ویژگی برای هر گره و یال ساخته شود.یکی از راه حل‌های موجود برای حل این مسئله مهندسی دستی[۲] که در آن دامنه خاص براساس ویژگی‌ها براساس دانش تخصصی ایجاد می‌شود است.

یک رویکرد جایگزین یادگیری نماینده‌های جایگزین با استفاده از حل یک مشکل بهینه سازی است.



شکل ۰-۱: گراف اولیه و گراف با یال های پیش بینی شده

۲-۴- الگوریتم node2vec

الگوریتم نیمه نظارت شـده[۳] node2vec برای اسـتخراج ویژگی‌های مقیاس پذیر در شـبکه‌ها اسـتفاده می‌شود. یک تابع هدف مبتنی بر گراف با استفاده از SGD[۴] براساس کار قبلی انجام شده در زمینه پردازش زبان طبیعی بهینه می‌شـود.این رویکرد ویژگی‌هایی را که بیشـترین احتمال ایجاد همسـایگی[۵] در یک گراف وجود دارد را نشان می‌دهد. با استفاده از یک رویکرد پیاده‌روی تصادفی دوم[۶] برای گره‌ها در گراف همسایگی مد نظر ایجاد می شود. سهم کلیدی این کار تعریف مفهوم انعطاف پذیر از گره‌های شبکه همسایگی ا ست. با

---

[۱] Machine learning

[۲] Hand engineering

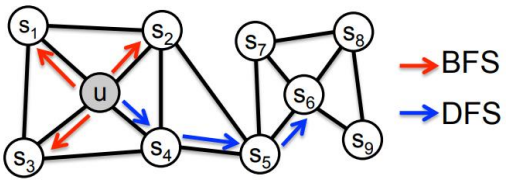
[۳] Semi-supervised

[۴] Stochastic gradient descent

[۵] Neighborhood

[۶] 2nd order random walk approach

انتخاب یک مفهوم درست از یک همسایگی، node2vec می‌تواند بازنمایی‌هایی که نقش گره‌ها در شبکه یا اجتماعی که به آن متعلق است را تشکیل می‌دهد یاد بگیرد. با ایجاد یک خانواده مسیر تصادفی مغرضانه[1] که به طور بسیار اثربخشی همسایگی‌های داده شده را جستوجو می‌کند.الگوریتم به دست‌آمده منعطف است و امکان کنترل در فضای جستوجو از طریق پارامترهای قابل تنظیم را به وجود می‌آورد. این روش می‌تواند طیف کامل معادلات مشاهده شده در شبکه را مدل کند .پارامترهای حاکم براین راهبرد جستوجو تفسیری بصری دارند و مسیر را به سمت راهبردهای کشف شبکه‌های مختلف سوق می‌دهد. این پارامترها نیز می توانند به طور مستقیم با استفاده از بخش کوچکی از داده های برچسب گذاری شده به صورت نیمه نظارت شده یاد گرفته شوند.هم‌چنین نشان داده می شود که چگونه بازنمایی گره‌های تکی به جفت گره‌ها گسترش پیدا می‌کند. به منظور تولید نمایش ویژگی گره‌ها، ویژگی آموخته شده ساخته می‌شوند. این ترکیب بندی به node2vec در وظایف پیش‌بینی کمک می‌کند . به عنوان مثال در شکل ۲-۲ گره‌های u و s_1 که هر دو عضو یک مجموعه محکم گره‌خورده هستند در حالی که u و s_6 عضو دو جامعه مختلف‌اند که یک نقش ساختاری یک گره هاب[2] را به اشتراک می‌گذارند که شبکه های دنیای واقعی معمولا مخلوطی از این معادلات را نشان می دهند[17].



شکل ۲-۱۰: راهبردهای جستوجو BFS و DFS

۲-۵- نتیجه‌گیری

شبکه ای که از دستگاه‌ها در اینترنت اشیا ایجاد می شود و آن را شبکه اینترنت اجتماعی اشیا نام می‌بریم.دستگاه‌ها در این شبکه برای مبادله خدمات،سرویس و اطلاعات با هم در ارتباط هستند.دستگاه‌ها در این شبکه را گره‌های یک گراف و ارتباط میان آن‌ها را یال‌های این گراف در نظر می‌گیریم.نیاز روزافزون به خدمات و اطلاعات بیشتر موضوع استفاده این دستگاه‌ها از خدمات و اطلاعات همدیگر را مطرح می‌کند.بنابراین از روش پیش‌بینی پیوندهای آینده برای توصیه کردن دستگاه و خدمات آن‌ها به یکدیگر استفاده می‌کنیم.در ادامه براساس موقعیت مکانی و مدت زمان باقی ماندن دستگاه‌های پویا اینترنت اشیا در

[1] Biased random walks

[2] Hub

یک مکان شبکه ی اینترنت اشیایی ایجاد می‌کنیم و سپس با استفاده از روش یادگیری ماشین node2vec برای پیش‌بینی دستگاه‌هایی که می‌توانند با هم ارتباط برقرار کنند استفاده می‌کنیم.

فصل ۳

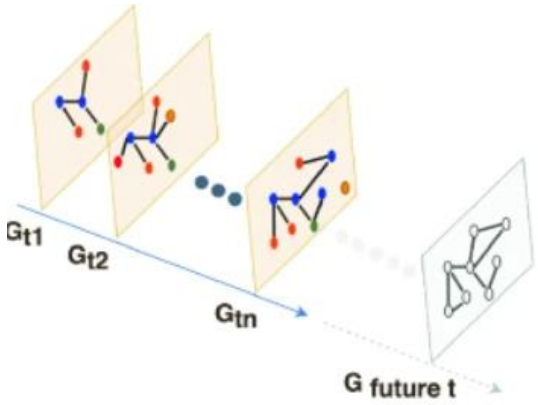
روش

۳-۱-مقدمه

در این بخش گراف پویا دستگاه‌های ناهمگون اینترنت اجتماعی اشیا را معرفی می‌کنیم.سپس جزئیات چارچوب این سیستم پیش‌بینی اینترنت اشیا را معرفی می‌کنیم.

۳-۲- مدل سازی گراف پویا اینترنت اجتماعی اشیا

یک گراف ناهگون پویا اینترنت اجتماعی اشیا مجموعه ای از گره و یال‌ها است که گره‌ها بیان‌گر د ستگاه‌های اینترنت ا شیا و و یال‌ها ن شان‌دهنده روابط مختلف ه ستند.تعریف ر سمی آن به شرح پیش‌رو است.شبکه اینترنت اجتماعی اشیا را می‌توان به عنوان یک توالی زمانی شبکه‌ها در نظر گرفت که در تصویر ۳-۱ نشان داده شده است[1].



شکل ۰-۱: مدل سازی شبکه ناهمگون اینترنت
اجتماعی اشیا

ابتدا $G = \{G^{T1}, G^{T2}, \ldots\ldots, G^{T-n}\}$ را داریم که G^t که تصـویر لحظه ای[۱] شـبکه در زمان t اسـت که $t \in \mathbb{Z}$. هر تصویر لحظه ای در شبکه شامل تعدادی گره و یال است به این‌صورت که $G^T = \{V^t, E^t, X^t\}$ که مجموعه‌ی گره‌ها $E^t = \{e^t_1, \ldots e^t_n\}$ شـامل N گره مشـاهده شـده در در زمان t اسـت و V^t مجموعه ی رئوسی است که در حداقل یک تا t گره به این‌صورت که $V^{t-1} \subseteq V^t$ مشارکت داشته‌اند.

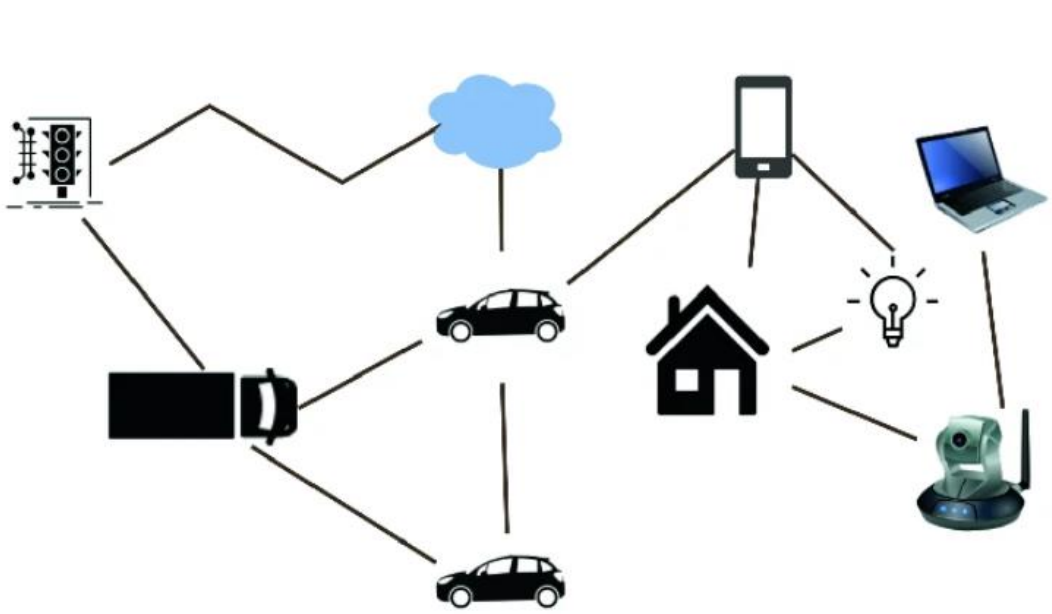
هر گره $e^t_n = (v^t_i, v^t_j)$ یک تاپل[۲] شامل دو گره که با هم در ارتباطند ا ست و ماتریس ویژگی در تایم t به صورت $X = [x^t_1, x^t_2, \ldots, x^t_i]$ که x_i بردار صفت[۳] گره v_n است.

شکل ۲-۳ مثالی از یک شبکه ناهمگون اینترنت ا شیا ا ست. برای این مثال کاربردی فرض می‌کنیم که یک گراف ناهمگون اینترنت اجتماعی ا شیا در زمان t از اطلاعات داده شده به د ست آمده ا ست.با توجه به این داده‌ها احتمال ایجاد یک رابطه(یال) بین هر دو دسـتگاه(گره) اینترنت اجتماعی اشـیا را پیش‌بینی می‌کنیم[1].

[۱] Snapshot

[۲] Tuple

[۳] Attribute vector



شکل ۰-۲: مثالی از یک شبکه ناهمگون اینترنت
اجتماعی اشیا

۳-۳- چارچوب پیشنهادی سیستم پیش بینی در اینترنت اجتماعی اشیا

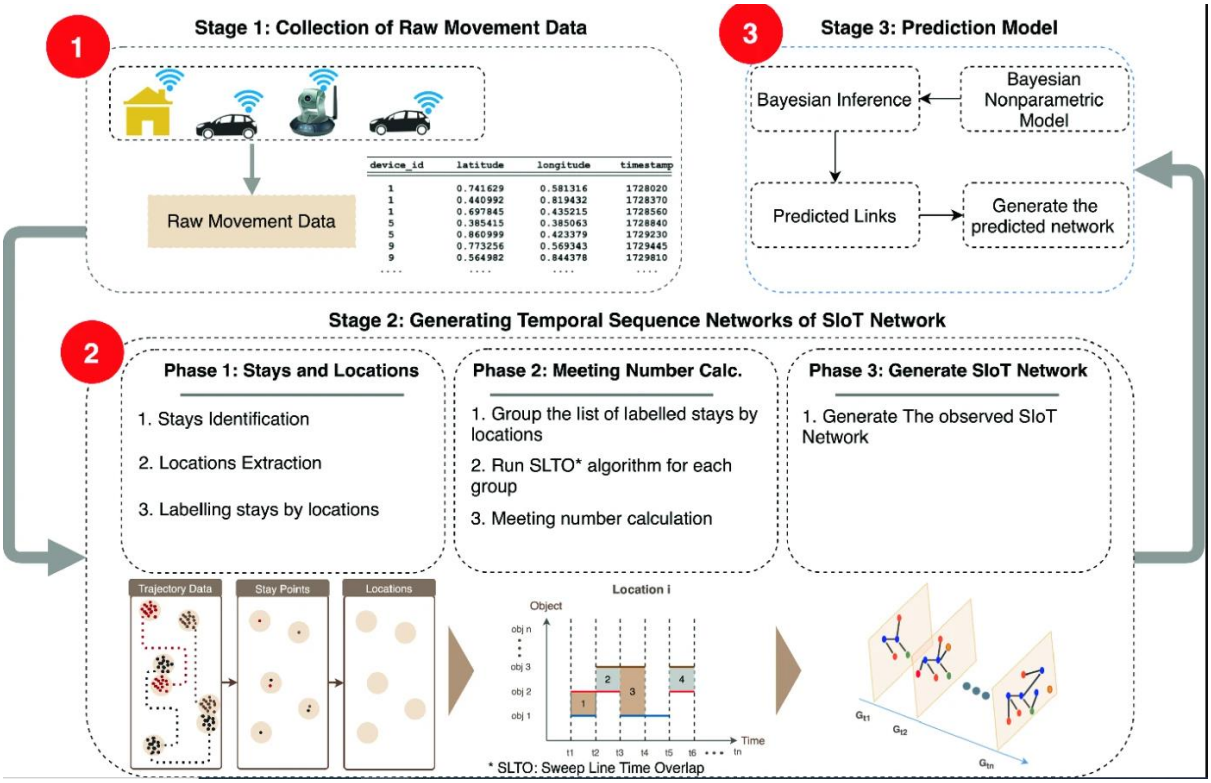
در این بخش چارچوب پیشنهادی برای سیستم پیشنهاد دهنده در شبکه اینترنت اجتماعی اشیا را توصیف می‌کنیم. شکل ۳-۳ بررسی اجمالی این چارچوب را نشان می‌دهد.این چار چوب شامل سه مرحله اصلی است:

مرحله اول: جمع آوری داده‌های خام حرکت دستگاه‌های اینترنت اشیا

مرحله دوم:تولید شبکه توالی زمانی اینترنت اجتماعی اشیا

مرحله سوم:پیش‌بینی روابط آینده اینترنت اجتماعی اشیا

در ادامه جزئیات بیشتری از هر کدام از این مراحل را بررسی می‌کنیم[1].



شکل ۰-۳: بررسی اجمالی چارچوب پیش‌بینی در
شبکه اینترنت اجتماعی اشیا

۳-۳-۱- جمع آوری داده‌های خام حرکت دستگاه‌های اینترنت اشیا

در مرحله اول این چارچوب، داده‌های خام حرکت دستگاه‌ها را جمع آوری می‌کنیم. دستگاه‌های اینترنت اشیا را به دو دسته کلی پویا و ایستا دسته بندی می‌کنیم. مختصات دستگاه‌های ایستا(مانند یک تیر چراغ برق) ثابت و مشخص هستند در حالی که مختصات دستگاه‌های پویا(مانند یک اتوبوس) پویا هستند و با حرکت کردن دستگاه تغییر می‌کنند.فرض می‌کنیم دستگاه‌های پویا مجهز به سیستم موقعیت یاب[۱] باشند و در هر برچسب زمانی[۲] مختصات این دستگاه‌ها را ارسال می شود . هم‌چنین فرض می‌کنیم دستگاه‌های

[۱] GPS

[۲] Timestamp

اینترنت ا شیا سابقه[۱] خود را به طور مداوم ار سال می‌کنند(به طور مثال هر ۶۰ ثانیه). هر سابقه شامل یک سری اطلاعات مهم به این صورت (شناسه دستگاه[۲]،عرض جغرافیایی[۳]،طول جغرافیایی[۴]،شروع برچسب زمانی[۵]،پایان برچسب زمانی[۶]) است[1].

هر سابقه تاریخچه موقعیت مکانی[۷] نشان دهنده‌ی یک نقطه روی کره زمین(طول و عرض جغرافیایی) و برچسب زمانی ا ست.این سابقه نشان می‌دهد یک د ستگاه در یک زمان خاص در چه موقعیت مکانی‌ای بوده است(نشان داده شده در شکل ۴-۳).

یک خط سیر[۸] دنباله ای از سابقه تاریخچه موقعیت مکانی یک دستگاه اینترنت اشیا است.

---

[۱] Record

[۲] Id device

[۳] Latitude

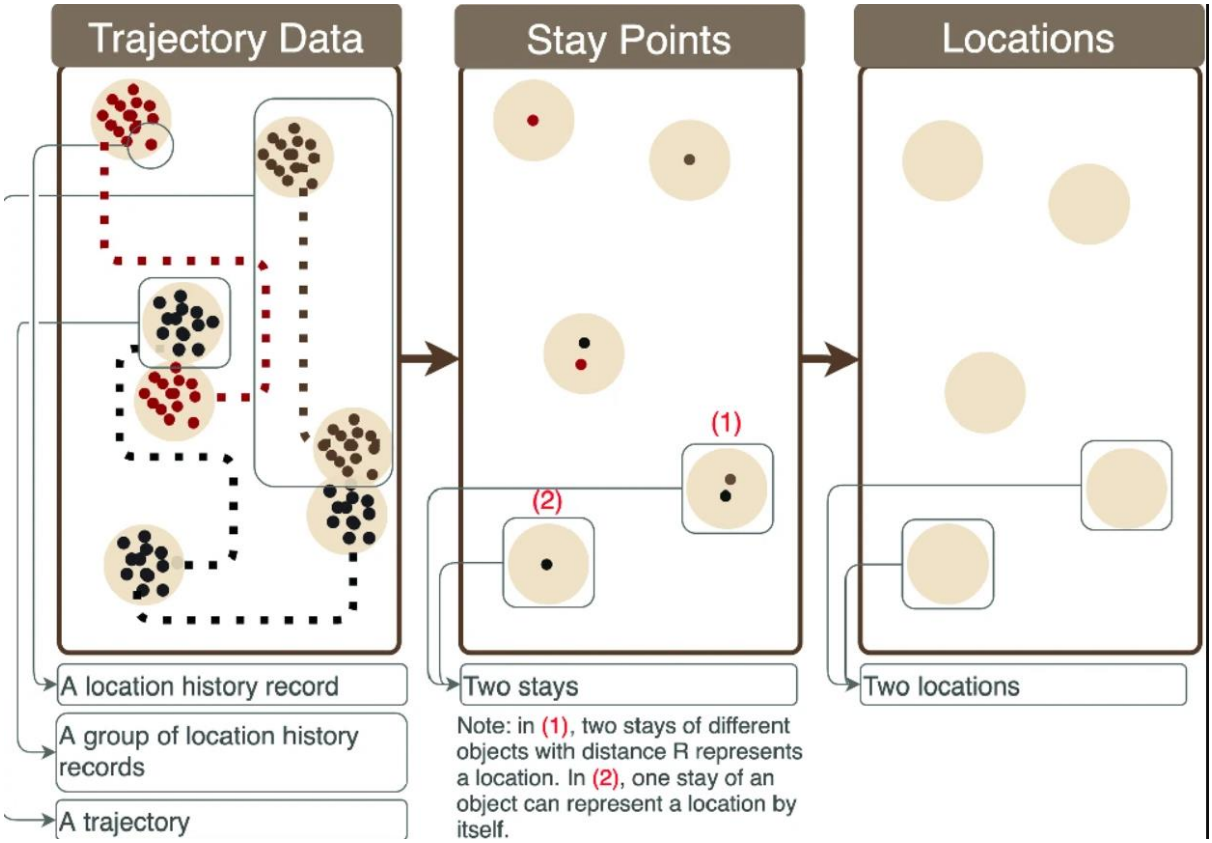
[۴] Longitude

[۵] Timestamp start

[۶] Timestamp stop

[۷] Location history record

[۸] Trajectory



شکل ۰-۴: تصویری از مکان، نقطه ماندن و مسیر
حرکت اینترنت اشیا

۳-۳-۲- تولید شبکه توالی زمانی اینترنت اجتماعی اشیا

در مرحله دوم براساس داده‌های خام حرکت دستگاه‌های اینترنت اشیا مشاهده شده در مرحله اول شبکه‌های اینترنت اجتماعی اشیا را ایجاد می‌کنیم . ایجاد شبکه توالی زمانی اینترنت اجتماعی اشیا در سه بخش انجام می‌پذیرد. بخش اول شامل یافتن نقطه ماندن[۱] با استفاده از داده‌های خام ،استخراج مکان و برچسب گذاری هر نقطه ماندن براساس مکان آن است. بخش دوم محاسبه تعداد ملاقات‌های میان دستگاه‌های اینترنت اشیا با استفاده الگوریتم هم‌پوشانی خط زمانی رفت و برگشت[۲] است. بخش آخر هم

[۱] Stay point

[۲] Sweep line time overlap(SLOT)

تولید شبکه توالی زمانی اینترنت اجتماعی اشیا است.در ادامه هر کدام از این بخش‌ها را با جزئیات بیشتر بررسی می‌کنیم[1].

بخش اول) یافتن نقطه ماندن با استفاده از داده‌های خام ،استخراج مکان و برچسب گذاری هر نقطه ماندن براساس مکان آن است. در این بخش می‌خواهیم بدانیم دستگاه‌ها در چه مکانی با یکدیگر ملاقات داشته اند. بنابراین ابتدا باید براساس داده‌های خام حرکت دستگاه‌ها نقطه ماندن برای هر دستگاه محاسبه کنیم.سپس مکان‌ها را براساس نقاط ماندن استخراج می‌کنیم. با این کار می‌توانیم بفهمیم هر دستگاه اینترنت اشیا در چه زمانی و چه مکانی بوده است. نقطه ماندن در واقع دنباله ای از N سابقه تاریخچه مکانی که به صورت طول و عرض جغرافیایی و زمان شروع و پایان نشان داده می‌شود است. طول و عرض جغرافیایی نشان دهنده میانگین طول و عرض جغرافیایی در دنباله هستند. زمان شروع بیان‌گر کوچک‌ترین برچسب زمانی و زمان پایان بیان‌گر بزرگ‌ترین برچسب زمانی در دنباله است[1].

یک مکان می‌تواند نشان‌دهنده طول و عرض جغرافیایی یک نقطه ماندن یا میانگین طول و عرض جغرافیایی یک گروه از نقاط ماندن باشد. یک گروه نقاط ماندن براساس فاصله‌شان دسته بندی می‌شوند که ما این اندازه را حداکثر ۲,۵ کیلومتر در نظر می‌گیریم. از الگوریتم شماره یک(شکل ۵-۳) برای یافتن نقاط ماندن، استخراج مکان‌ها و برچسب گذاری نقاط ماندن براساس مکان‌های استخراج شده استفاده می‌کنیم. ورودی این الگوریتم داده‌های خام حرکت دستگاه‌ها در اینترنت اشیا است. خروجی این الگوریتم لیستی از نقاط ماندن که براساس مکان برچسب گذاری شده اند است. به دلیل اینکه در این الگوریتم مسافت بین هر داده‌ی خام مشاهده شده محاسبه می‌شود مرتبه اجرایی[۱] این الگوریتم از درجه دوم است[1].

۳-۴- یافتن نقاط ماندن و استخراج مکان‌ها

در قدم اول می‌خواهیم نقاط ماندن را شناسایی کنیم . در این قدم نقطه ماندن برای هر دستگاه اینترنت اشیا براساس داده‌های خام حرکت آن دستگاه را شناسایی می‌کنیم ما از داده‌های خام مجموعه داده[۲] [] که براساس داده‌های دستگاه‌های واقعی اینترنت اشیا است استفاده می‌کنیم. ابتدا بازه‌زمانی[۳] یک نقطه ماندن رو مشخص می‌کنیم به عنوان مثال در پیاده سازی این الگوریتم مقدار آن براساس ماتریس‌های

[۱] Time complexity

[۲] Data set

[۳] Time period

مجاورت[1] موجود[] ۱۰ است به این معنا که باید دستگاه‌هایی که ۱۰ دقیقه در آن مکان استقرار داشته اند را شناسایی کنیم.

```
Input  : The raw movement data of IoT objects
Output: Stays Labelled with extracted locations
data = raw movement data;
StayPeriod = N;
ASSIGN EMPTY LIST to Stay;
ASSIGN EMPTY LIST to IdentifiedStays;
ASSIGN EMPTY LIST to Locations;
for i ← 1 to data.length − 1 do
    Stay.append(data[i]);
    for j ← i + 1 to data.length do
        distance ← distance(data[i], data[j]) ;
        if (distance <= R) then
            Stay[i].append(data[j]);
        else
            i ← j;
            Break;
        end
    end
end
```

شکل ۰-۵: الگوریتم یافتن نقطه ماندن

برا ساس فر ضیات، هر سابقه داده‌های خام حرکت هر یک دقیقه یک‌بار ار سال می شود بنابراین ده سابقه نشان‌دهنده ی ده دقیقه است. با استفاده از رابطه (۳-۱) فاصله بین سابقه‌ی داده‌ی خام حرکت دستگاه‌ها را محاسبه می‌کنیم، d نشان‌دهنده ی فاصله‌ی بین دو سابقه‌ی تاریخچه موقعیت مکانی، r بیان‌گر شعاع کره زمین، θ_1 θ_2 عرض جغرافیایی دو سابقه تاریخچه موقعیت مکانی و $\lambda_1\lambda_2$ طول جغرافیایی دو سابقه تاریخچه موقعیت مکانی اند.

$$distance = 2r . \sin^{-1}\left(\sqrt{sin^2\left(\frac{(\theta_2 - \theta_1)}{2}\right) + \cos(\theta_1).\cos(\theta_2).sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

رابطه ۰-۱

[1] Adjacency matrices

سپس الگوریتم براساس این رابطه اگر فاصله آن‌ها کمتر یا مساوی آستانه R (۲,۵ کیلومتر) باشد آن‌ها را در لیست نقاط ماندن شناخته شده[۱] گروه بندی می‌کند.

در ادامه(شـــکل ۶-۳) الگوریتم هر گروه را بررســـی می‌کند اگر هر کدام از این گروه‌ها ســـوابقی بزرگ‌تر یا مساوی زمان ماندن[۲] که براساس ماتریس همسایگی آن را ۳ در نظر می‌گیریم داشته باشند الگوریتم میانگین طول و عرض جغرافیایی گروه را در نظر می‌گیرد. هم‌چنین کوچک‌ترین برچســـب زمانی گروه را به عنوان زمان شروع ماندن و بزرگ‌ترین برچسب زمانی گروه را به عنوان زمان پایان ماندن در نظر می‌گیرد.

```
18 for i ← 1 to Stay.length do
19     if Stay[i].length >= StayPeriod then
20         objID ← Stay[i].objID;
21         lat ← Compute avg lat of the stay;
22         long ← Compute avg long of the stay;
23         Start-Time ← min(Stay[i].timestamp);
24         End-Time ← max(Stay[i].timestamp);
25         IdentifiedStays.append([objID,lat,long,Start-Time,End-Time]);
26     end
27 end
```

شکل ۰-۶: الگوریتم یافتن نقاط ماندن شناسایی شده

در ادامه به اســـتخراج مکان‌ها می‌پردازیم(شـــکل ۷-۳).الگوریتم لیســـتی از مکان‌ها را براســـاس نقاط ماندن شـــناســـایی شـــده اســـتخراج می‌کند.از آنجایی که نقاط ماندن با طول و عرض جغرافیایی نشـــان داده می شوند الگوریتم فا صله بین این نقاط ماندن را محا سبه می‌کند و برا ساس حد آ ستانه R آن‌ها را د سته بندی می‌کند. الگوریتم از میانگین طول و عرض جغرافیایی این نقاط ماندن برای نشـــان دادن یک موقعیت مکانی استفاده می‌کند. اگر یک نقطه ماندن وجود داشته باشد که با نقاط ماندن دیگر گروه بندی نشده باشد، آن نقطه ماندن می‌تواند به تنهایی بیان‌گر یک موقعیت مکانی باشـــد. در مرحله آخر نقاط ماندن را براســـاس موقعیت مکانی برچسب گذاری می‌کنیم.نقاط شناسایی شده را براساس یکی از نقاط استخراج شده برچسب گذاری می‌کنیم. پس از پیاده ســـازی این الگوریتم با اســـتفاده از اطلاعات موجود در [] ، تعداد ملاقات های گزارش شده ۱۲۵۰۰۰ ملاقات میان دستگاه‌ها است.

[۱] Identified Stays

[۲] Stay period

```
28 for i ← 1 to IdentifiedStays.length − 1 do
29 |  coords=[ ];
30 |  coords.append([IdentifiedStays[i].lat,IdentifiedStays[i].long]);
31 |  for j ← i + 1 to IdentifiedStays.length do
32 |  |  distance ← distance(IdentifiedStays[i], IdentifiedStays[j]) ;
33 |  |  if (distance <= R) then
34 |  |  |  coords.append([IdentifiedStays[j].lat,IdentifiedStays[j].long]);
35 |  |  end
36 |  end
37 |  lat ← Compute avg lat of coords;
38 |  long ← Compute avg long coords;
39 |  Locations.append([lat,long]);
40 end
```

شکل ۰-۶: الگوریتم یافتن نقاط ماندن شناسایی شده

۳-۵- محاسبه تعداد ملاقات‌های میان دستگاه‌ها

با استفاده الگوریتم هم‌پوشانی خط زمانی رفت و برگشت[1] تعداد ملاقات‌های میان دستگاه‌ها را محا سبه می‌کنیم.الگوریتم شماره ۲ (شکل ۳-۷) برای ت شخیص و گزارش تمامی دوره‌های هم‌پو شانی در لیست‌های ماندن که توسط الگوریتم شماره ۱ به دست‌آمده اند ایجاد می‌کنیم. هدف این الگوریتم مشخص کردن هر دو د ستگاه اینترنت ا شیا در یک موقعیت مکانی م شخص ملاقات دا شته اند ا ست. این الگوریتم با ایده گرفتن از الگوریتم خط رفت و برگ شت که در علم هند سه برای یافتن تقاطع بین گروهی از پاره خط‌ها ا ستفاده می شود ایجاد شده ا ست. روش کار این الگوریتم به این صورت ا ست که یک خط مجازی رفت و برگ شت بر روی محور طول‌ها از چپ به را ست برای یافتن وقفه‌ها روی محور عرض‌ها حرکت می‌کند. وقتی این خط رفت و برگشت یک هم‌پوشانی میان ماندن‌ها پیدا کند آن‌ها را گزارش می‌کند. این الگوریتم در مرحله اصلی دارد .

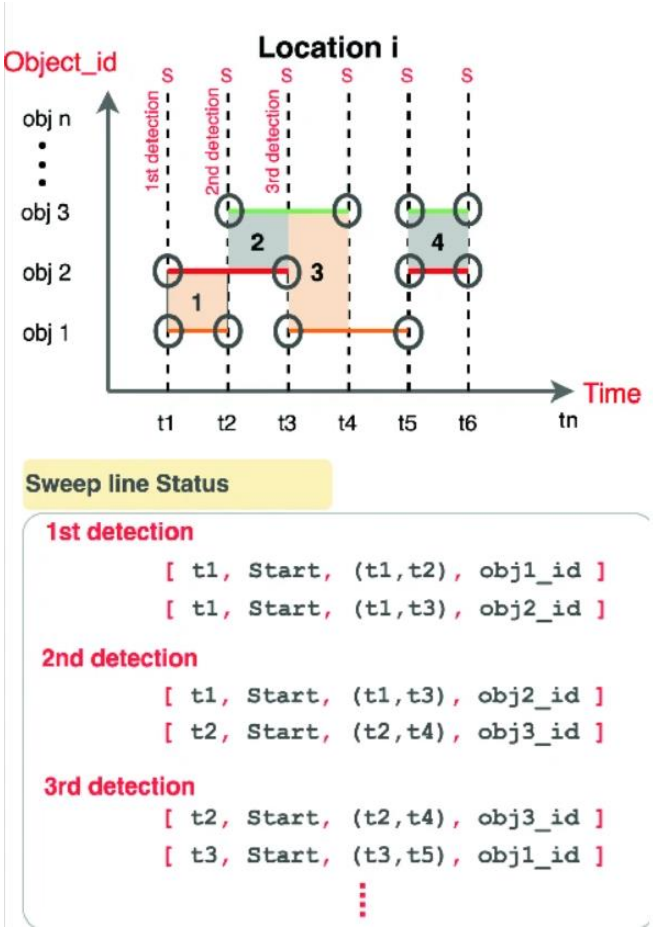
ابتدا تمام وقفه‌های ماندن‌ها ذخیر سازی می شوند.چون هدف الگوریتم یافتن دوره‌های هم‌پو شانی میان مجموعه‌ای از ماندن‌ها است دو ساختار اولیه ،یک صف اولویت S برای ذخیره‌سازی تمامی وقفه‌های مرتب سازی شده که از الگوریتم ۱ به دست آمده و یک وضعیت خط رفت و برگشتQ برای بررسی ماندن‌ها از چپ به راست تعریف می‌شوند. قدم بعدی اجرای خط رفت و برگشت است.

[1] Sweep line time overlap

```
Input  : A list of stays
Output: A list of pair of objects with the
        overlapped period among them
1  begin
2      Q = The list of stays from Algorithm 1
        as [[stay endpoint value, status,
        interval, objeID]];
3      S = Initialize the Sweep line for storing
        detected stays;
4      while Q is not empty do
5          currentEvent ← Q.get();
6          if (currentEvent.Status == Start)
            then
7              - Insert currentEvent into S;
8          else
9              - CheckOverlap(S ,
                currentEvent);
10             - Delete currentEvent from S;
```

شکل ۰-۷: الگوریتم محاسبه ملاقات میان دستگاه‌ها

در ادامه خط رفت و برگشت S (خط ۴ تا ۱۰ در شکل ۳-۷)را اجرا می‌کنیم. نقاط انتهایی بازه تک به تک از Q دریافت می شوند تا به خط رفت و برگشت S اجازه برر سی آن‌ها داده شود.خط رفت و برگشت شروع ماندن را در فضا تشخیص می‌دهد و آن را به S اضافه می‌کند. هم‌چنین وقتی پایان اقامت را تشخیص می‌دهد الگوریتم آخرین مقدار موجود در S را برر سی می‌کند. اگر این مقدار شروع آن اقامت بود بدون هیچ د ستور خاص دیگری اقامت را از S حذف می‌کند.اگر آخرین مقدار در S مقدار پایانی اقامت به اتمام ر سیده نبود آن‌گاه یک یا چند هم‌پوشانی میان این ملاقات و ملاقات‌های دیگر فعال در S تشخیص داده می‌شود. چون زمان ملاقات فقط زمانی محا سبه می شود که هم‌پو شانی بین د ستگاه‌ها وجود دا شته با شد شکل(۳-۸)پیچیدگی زمانی این الگوریتم $O(NlogN + l)$ است .



شکل ۰-۸: مثالی از روش کار الگوریتم شماره ۲ ،
اولین تشخیص در زمان t1 اتفاق افتاده است.

الگوریتم شماره ۳ هم‌پوشانی‌های کشف شده توسط الگوریتم هم‌پوشانی خط زمانی رفت و برگشت(الگوریتم شماره ۲) را گزارش می‌کند این الگوریتم تعداد هم‌پوشانی های تشخیص داده شده را محاسبه و گزارش می‌کند.

Algorithm 3: CheckOverlap

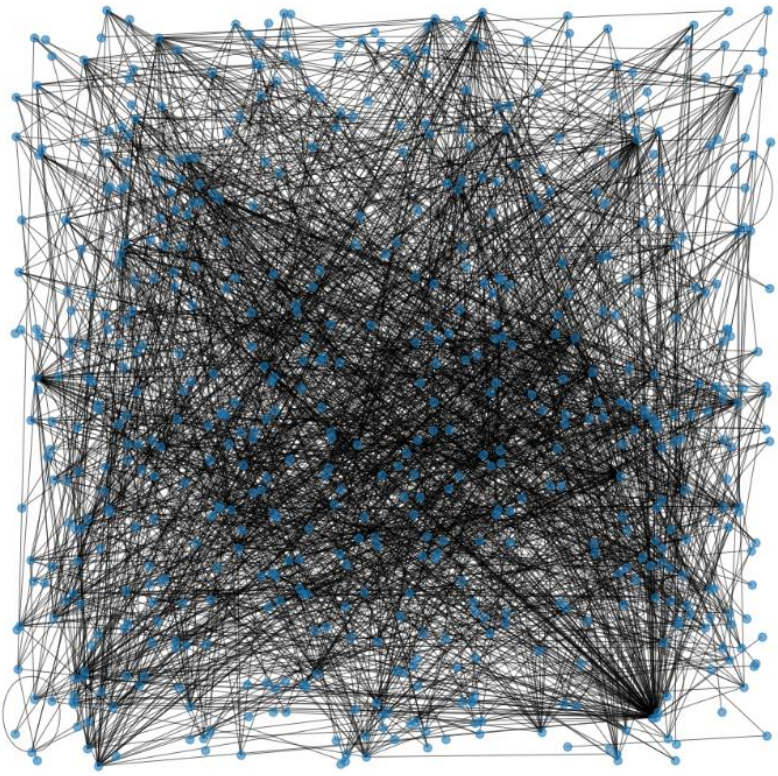
```
Input  : S, currentEvent
Output: Report the overlapping among stays
1 begin
2     L2 = currentEvent
3     for (L1 in S) do
4         if L1.start < L2.end & (L2.start < L1.end) then
5             OverlapPeriod ← min(L1.end, L2.end) − max(L1.start, L2.start)
6             - Report the overlap (ids of objects and OverlapPeriod)
```

شکل ۰-۹: الگوریتم بررسی هم‌پوشانی‌ها

پس از اجرای الگوریتم‌های ۲،۱ و ۳ و ایجاد جدول(جدول۳-۱) از جفت گره‌های متصل گراف (شکل۳-۱۰) شبکه ایجاد می‌شود.

جدول ۳-۱:نمونه ای از جدول ایجاد شده از دستگاه‌ها و ارتباط آن‌ها(این جدول فقط تعداد کمی از سوابق را نشان می‌دهد و جدول اصلی سوابق بیشتری دارد)

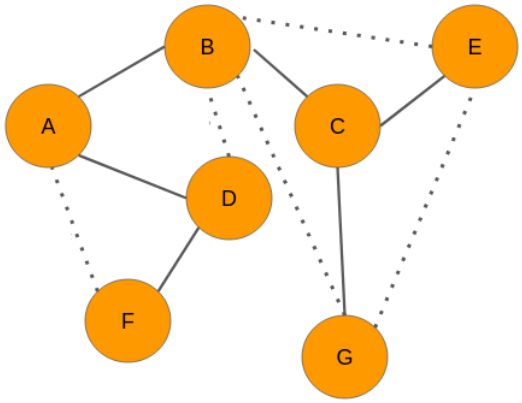
گره ۱	گره ۲
۱۶۱۷۳	۱۶۲۰۵
۱۶۱۷۹	۱۶۲۱۳
۱۶۱۷۱	۱۶۱۸۷
۱۶۲۰۱	۱۶۱۸۱
۱۶۱۵۹	۱۶۱۹۷



شکل ۰-۱۰: گراف ایجاد شده از شبکه اینترنت اجتماعی اشیا

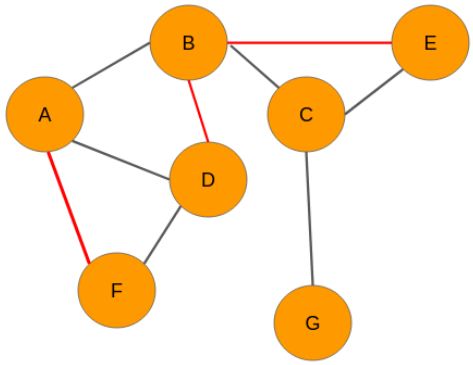
۳-۶- پیش‌بینی پیوندهای آینده

در ادامه پیاده‌سازی الگوریتم شماره ۳ خروجی به صورت یک جدول دو ستونه که سطرهای آن شناسه شناسایی دستگاه‌هایی که با هم مرتبط هستند است. اگر بتوانیم یک گراف را به صورت یک مجموعه‌داده ویژگی‌های آن ن شان دهیم آن‌گاه می‌توانیم با ا ستفاده از یادگیری ما شین[۱] پیوندهای احتمالی بین گره‌های غیرمتصل را در یک گراف پیش‌بینی کنیم.روش کلی که از آن استفاده می‌کنیم را در ادامه مرحله به مرحله بررسی می‌کنیم.گراف (شکل ۳-۱۰) را به عنوان نمونه در نظر می‌گیریم . در این گراف ۷ گره وجود دارد که گره‌های متصل به هم با خط و گره‌های غیر متصل با خط چین نشان داده شده‌اند.



شکل ۰-۱۱: گراف نمونه در زمان t

حالا در ادامه فرض می‌کنیم یک سری تحلیل انجام شده و گراف(شکل ۳-۱۲) به دست‌آمده است که یال‌های قرمز نشان‌دهنده پیوندهای جدید ایجاد شده اند.تعدادی پیوند جدید ایجاد شده است.



شکل ۰-۱۲: گراف نمونه در زمان t+n

[۱] Machine learning

برای ایجاد یک مدل یادگیری ماشـــین به تعدادی متغیر پیش‌بینی کننده[1] و یک متغیر هدف[2] نیاز داریم . هدف ما پیش‌بینی احتمال وجود پیوند بین دو گره غیر متصـل اسـت. از گراف در زمان t (شـکل ۳-۱۱) چند گره غیرمتصل(متغیرهای پیش بینی‌کننده) را ا ستخراج می‌کنیم ، گره‌هایی را انتخاب کرده‌ایم که فقط چند یال از هم فاصله دارند گره A-F ,B-D,B-E,B-G و E-G گره‌های مد نظر هستند. قدم بعدی ایجاد ویژگی برای هر جفت گره اســـت اما متغیر هدف هنوز مشـــخص نشـــده اســـت بنابراین گراف را در زمان t+n (شکل ۳-۱۲) بررسی می‌کنیم که سه جفت گره متصل جدید در آن به وجود آمده است جفت A-F,B-D وB-E بنابراین در جدول ویژگی(جدول۳-۲) که ایجاد می‌کنیم به این پیوندهای ایجاد شـــده مقدار ۱ و به پیوندهایی که حدس زده ایم اما ایجاد نشدند مقدار ۰ را اختصاص می‌دهیم.

جدول ۳-۲: جدول ویژگی گره‌ها

نوع پیوند	ویژگی
۱	ویژگی جفت گره A-F
۱	ویژگی جفت گره B-D
۱	ویژگی جفت گره B-E
۰	ویژگی جفت گره B-G
۰	ویژگی جفت گره E-G

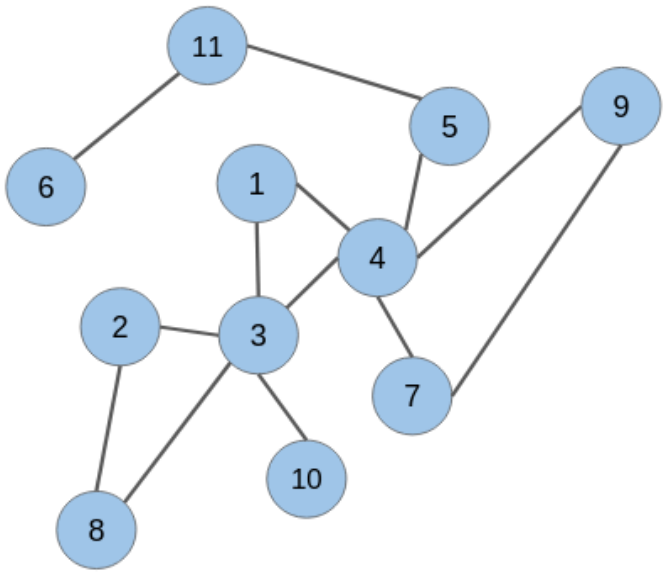
حال که با توجه به جدول متغیر هدف هم به دست آمد می‌توانیم با استفاده از داده‌هایی که از شبکه اینترنت اجتماعی اشـــیا به دســـت آورده‌ایم می‌توانیم مدل یادگیری ماشـــین برای پیش‌بینی پیوندهای آینده را پیاده‌ســـازی کنیم. در مثالی که بررســـی شـــد گراف را در زمان t و t+n در اختیار داشـــتم اما جفت گره‌هایی(دستگاه‌های اینترنت اجتماعی اشیا) فقط در زمان حال موجود هستند بنابراین برای حل این مشکل نیز باید یک راه‌حل ارائه دهیم.

[1] Predictor variables

[2] Target variable

۳-۷- ایجاد مدل

با توجه به اینکه گراف شـــبکه‌ای که می‌خواهیم پیوندهای آینده را برای آن پیش‌بینی کنیم را فقط در زمان حال داریم بنابراین باید مدلی ارائه دهیم که بتوان براســـاس آن و بدون گراف در زمان آینده متغیرهای هدف را برچسب گذاری کرد. گراف(شکل ۳-۱۳) را به عنوان نمونه برای پیاده سازی این مدل در نظر می‌گیریم که براساس شبکه‌ای که داریم گره‌ها دستگاه‌ها و یال‌ها همان ارتباط بین آن‌ها هستند.

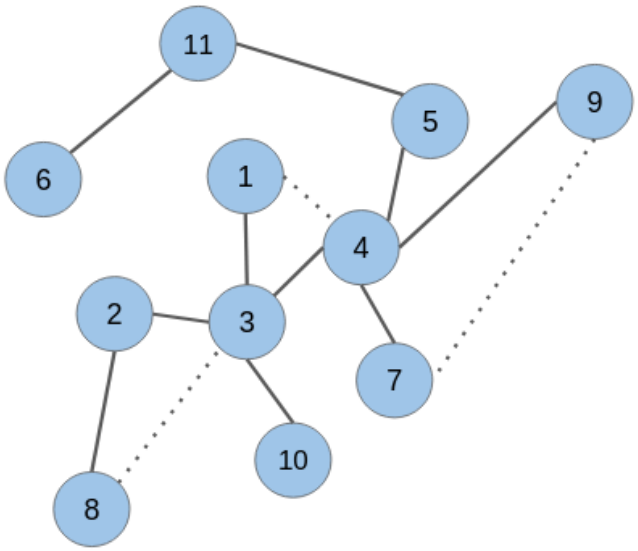


شکل ۰-۱۳: گراف نمونه برای پیاده‌سازی مدل در زمان حال

چند جفت گره نامزد که ممکن اســت در آینده بین‌ها ارتباطی شـکل بگیرد گره 1-2,2-4,5-6,8-10 و... هســـتند. مدلی که ارائه می‌دهیم پیش‌بینی می‌کند که پیوند بین این جفت گره‌ها ایجاد می‌شـــود یا نمی‌شـود. ابتدا باید یک مجموعه داده آموزشـی[۱] از این گراف به دسـت بیاوریم. برای انجام این‌کار شـبکه یا گراف را در زمان گذشته در نظر می‌گیریم که همان گراف با تعدادی یال کمتر است چون همان‌طور که قبل تر اشاره شد ارتباطات در این شبکه با گذشت زمان ساخته می‌شوند. بنابراین به سادگی می‌توانیم تعدادی از

[۱] Training dataset

یال‌های این گراف را به صورت تصادفی حذف کنیم و روشی که در بخش ۳-۶ توضیح دادیم را برای ایجاد یک مجموعه داده از آن اجرا کنیم. تعدادی از یال‌های گراف نمونه(شکل ۳-۱۳) را حذف می‌کنیم تا گراف جدیدی ایجاد شود(شکل ۳-۱۴) هنگام حذف یال‌ها باید از حذف یالی که باعث ایجاد یک گره بدون یال (ارتباط) می‌شود خودداری کنیم.



شکل ۰-۱۴: گراف نمونه که تعدادی از یال‌های آن حذف شده‌اند

یال‌هایی که با خط‌چین نشان داده شده‌اند یال بین جفت گره 9-7, 4-1 و 8-3 حذف شده‌اند. حال برای تمامی گره‌های غیرمتصل و گره‌هایی که حذف شده‌اند جدول ویژگی (جدول ۳-۳) را ایجاد می‌کنیم.

جدول ۳-۳ :جدول ویژگی گره‌ها

نوع پیوند	ویژگی
۰	ویژگی جفت گره 1-2
۰	ویژگی جفت گره1-5
۰	ویژگی جفت گره1-7
۰	ویژگی جفت گره1-8
۰	ویژگی جفت گره1-9
۰	ویژگی جفت گره1-10
۰	ویژگی جفت گره2-4

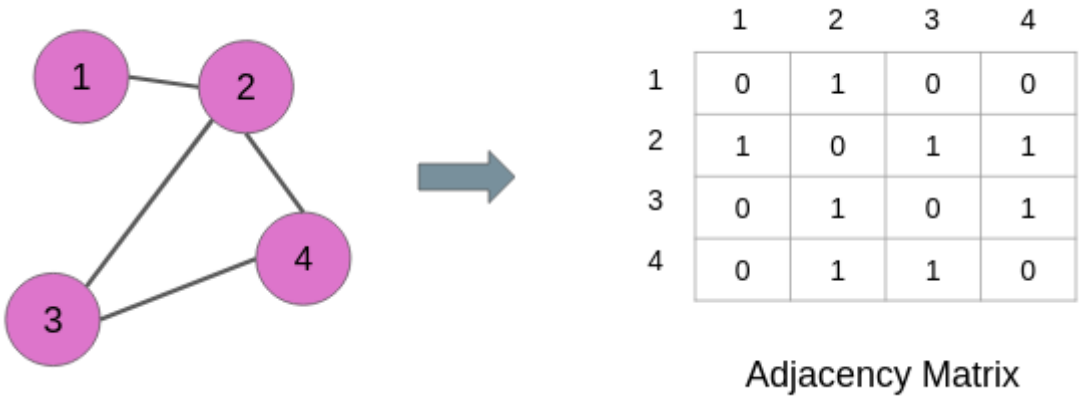
۰	ویژگی جفت گره2-10
۰	ویژگی جفت گره3-5
۰	ویژگی جفت گره3-7
۰	ویژگی جفت گره3-9
۰	ویژگی جفت گره4-8
۰	ویژگی جفت گره4-10
۰	ویژگی جفت گره4-11
۰	ویژگی جفت گره5-6
۰	ویژگی جفت گره5-7
۰	ویژگی جفت گره5-9
۰	ویژگی جفت گره8-10
۱	ویژگی جفت گره1-4
۱	ویژگی جفت گره3-8
۱	ویژگی جفت گره7-9

همان‌طور که در جدول مشخص است متغیر هدف به شدت نامتعادل[۱] است. که این مشکل در گراف شبکه اینترنت اشیا که ایجاد کرده‌ایم نیز وجود دارد.تعداد جفت گره‌های غیرمتصل زیاد است . این مشکل را با استفاده از یادگیری ماشین پایتون حل می‌کنیم.

تا این‌جا دانستیم باید براساس داده‌های موجود یک مجموعه داده ایجاد کنیم اما بخش زیادی از این مجموعه داده یا در واقع همین جدوالی که ایجاد کرده‌ایم نمونه‌های منفی یا همان جفت گره‌های غیر متصلند در ادامه باید جفت گره‌های غیرمتصل را استخراج و حذف کنیم.ابتدا باید یک ماتریس مجاورت برای یافتن اینکه کدام جفت گره‌ها متصل نیستند ایجاد کنیم به این‌صورت که سطر و ستون‌ها بیان‌گر گره‌ها هستند در شکل ۳-۱۵ یک گراف نمونه و ماتریس مجاورت آن نشان داده شده. اتصال یا عدم اتصال گره‌ها در این ماتریس با مقادیر ۰ و ۱ مشخص می‌شود ۱ به معنای این است که بین آن دو جفت گره اتصال برقرار است و ۰ به معنای این است که بین این دو گره اتصالی وجود ندارند. حال با استفاده از جدولی که از پیش از این

[۱] imbalanced

از شـــبکه اینترنت اجتماعی اشـــیا ایجاد کرده‌ایم و با اســـتفاده از دســـتورات موجود در زبان پایتون ماتریس مجاورت جدولی که موجود است را ایجاد می‌کنیم.



شکل ۰-۱۵: یک گراف نمونه و ماتریس مجاورت آن در سمت چپ

بدیهی اســـت که ماتریس مجاورتی که ایجاد می‌شـــود یک ماتریس مربعی اســـت . ماتریس مجاورت را برای یافتن مقادیر ۰ جســـت‌وجو می‌کنیم و هم‌چنین می‌دانیم که در این ماتریس مربعی مقادیر بالا و پایین قطر ماتریس برابرند پس کافی ا ست فقط بالا یا پایین قطر ماتریس را برر سی کنیم. جفت گره‌های غیرمت صل را در یک جدول دیگر ذخیره می‌کنیم.این جفت‌گره‌ها به عنوان نمونه منفی عمل می‌کنند و دلیل دخیره سازی آن‌ها نیز همین مسئله است.

پیش‌تر اشاره کردیم که تعدادی از گره‌ها را به صورت تصادفی حذف می‌کنیم اما ممکن است در این عملیات تعدادی از گره‌ها که ارتباط کمی دارند حذف شوند.بنابراین در پیاده سازی این موضوع را در نظر گرفته و در مرحله حذف کردن تمامی گره‌های گراف متصل باقی بمانند. قبل از حذف هر گره چک می‌کنیم که گره‌هایی که به آن متصل است بیشتر از ۱ باشد. و در نهایت گره‌های قابل حذف را در جدولی دیگر ذخیره می‌کنیم.

در ادامه جدول گره‌های قابل حذف را به جدول گره‌های غیر متصل اضافه می‌کنیم و از آن‌جایی که گره‌های قابل حذف نمونه‌های مثبت ما هســـتند به آن‌ها مقدار ۱ را می‌دهیم. ســـپس با اســـتفاده از الگوریتم node2vec جدول ویژگی برای تمامی گره‌ها پس از حذف تعدادی از گره‌ها را استخراج می‌کنیم.

۳-۸- کدهای پیش‌بینی پیوندهای آینده

برای ایجاد ماتریس مجاورت از قطعه کد زیر استفاده شده در خط اول ابتدا همه گره‌ها با هم ترکیب شده و در یک لیست ذخیره می شوند. سپس در خط بعدی تمامی موارد تکراری حذف می شوند و در نهایت در خط سوم ماتریس مجاورت را تشکیل می‌دهیم.

```
node_list = node_list_1 + node_list_2

node_list = list(dict.fromkeys(node_list))

adj_G = nx.to_numpy_matrix(G, nodelist = node_list)
```

شکل ۰-۱۶: قطعه کد برای تشکیل ماتریس مجاورت

در ادامه باید ماتریس ایجاد شده را برای یافتن مقادیر ۰ جست‌وجو کنیم(شکل ۳-۱۷) و همان‌طور که پیش تر نیز اشاره شد برای این کار هم می‌توان مقادیر بالای قطر یا پایین قطر را جست‌وجو کرد در قطعه کد زیر روش این کار نشان داده شده است. ابتدا یک لیست برای دریافت تمامی گره‌های غیرمتصل ایجاد می‌کنیم سپس ماتریس را جست و جو می‌کنیم و نتیجه جستجو را به انتهای لیست اضافه می‌کنیم. این جفت گره‌های غیر متصل که پیدا می‌شوند به عنوان نمونه منفی در مدل عمل می‌کنند.

```
all_unconnected_pairs = []
offset = 0
for i in tqdm(range(adj_G.shape[0])):
  for j in range(offset,adj_G.shape[1]):
    if i != j:
      if nx.shortest_path_length(G, str(i), str(j)) <=2:
        if adj_G[i,j] == 0:
          all_unconnected_pairs.append([node_list[i],node_list[j]])
```

شکل ۰-۱۷: کد جست‌وجوی ماتریس برای یافتن مقادیر ۰

نمونه‌های منفی یا جفت گره‌های غیر متصل را در یک جدول[۱] ذخیر می‌کنیم. در خط اول گره اول را در سطر اول یک لیست و در خط دوم گره دوم را در یک لیست دیگر ذخیره می‌کنیم و نهایتا این لیست‌ها در یک جدول ذخیره می‌کنیم.

```
node_1_unlinked = [i[0] for i in all_unconnected_pairs]
node_2_unlinked = [i[1] for i in all_unconnected_pairs]

data = pd.DataFrame({'node_1': node_1_unlinked,
                     'node_2': node_2_unlinked})
```

شکل ۰-۱۸: ذخیر جفت گره‌های غیرمتصل در یک جدول

حال باید تمامی یال‌هایی که می‌توان آن‌ها را به طور تصـــادفی حذف کرد را بیابیم. یک لیســـت خالی برای ذخیره این جفت گره‌های ایجاد می‌کنیم و سپس جفت‌گره را حذف می‌کنیم و در نهایت بررسی می‌کنیم که گراف از هم جدا نشده باشد و تعداد گره‌ها یکسان باشد.

[۱] Data frame

```
initial_node_count = len(G.nodes)

fb_df_temp = fb_df.copy()


omissible_links_index = []

for i in tqdm(fb_df.index.values):

G_temp = nx.from_pandas_edgelist(fb_df_temp.drop(index=i), "node_1", "node_2", create_using=nx.Graph())


if (nx.number_connected_components(G_temp) == 1) and (len(G_temp.nodes) == initial_node_count):
        omissible_links_index.append(i)
        fb_df_temp = fb_df_temp.drop(index=i)
```

شکل ۰-۱۹: یافتن گره‌های قابل حذف شدن

در ادامه گره‌های قابل حذف را به جدول گره‌های غیرمتصـــل الحاق می‌کنیم آن‌جایی که که این گره‌ها نمونه‌های مثبت هستند مقدار ۱ را به آن‌ها اختصاص می‌دهیم. ابتدا یک جدول از گره‌های قابل حذف ایجاد می‌کنیم سپس متغیر هدف ’link‘ را اضافه می‌کنیم.

```
fb_df_ghost = fb_df.loc[omissible_links_index]

fb_df_ghost['link'] = 1

data = data.append(fb_df_ghost[['node_1', 'node_2', 'link']], ignore_index=True)
```

شکل ۰-۲۰: ایجاد جدول link

۳-۸-۱-استخراج ویژگی

با اســـتفاده از الگوریتم node2vec ویژگی گره‌ها را بعد از حذف پیوندها اســـتخراج می‌کنیم.یک گراف جدید پس از حذف پیوندهای قابل حذف ایجاد می‌کنیم(شـکل ۳-۱۹).در ادامه کتابخانه node2vec

را نصب می‌کنیم و سپس مدل node2vec را به گراف‌مان آموزش می‌دهیم. مدل آموزش داده شده را روی تمام جفت گره‌های جدول data اجرا می‌کنیم.

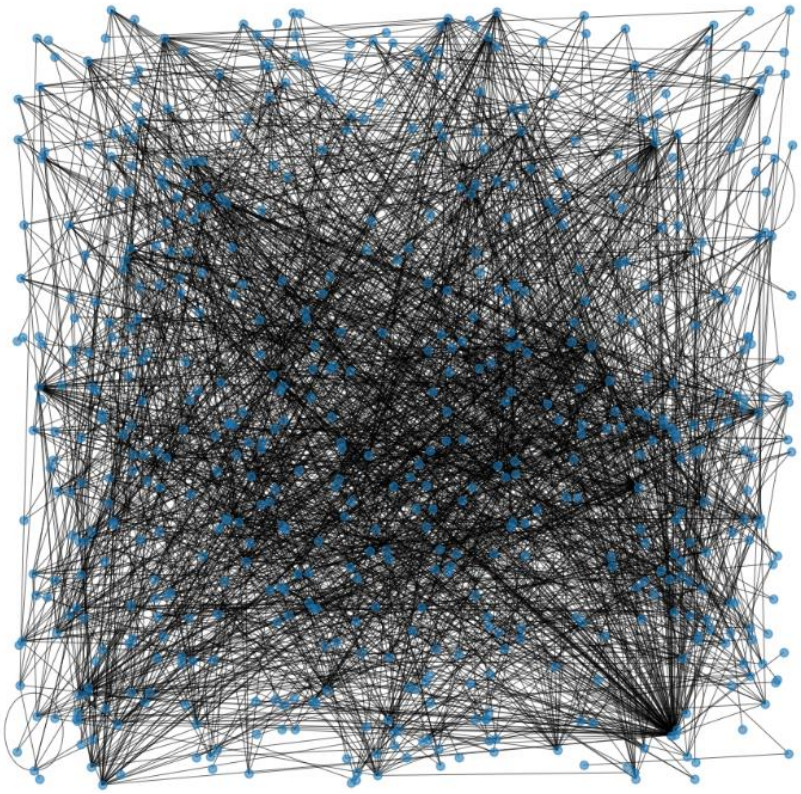
```
fb_df_partial = fb_df.drop(index=fb_df_ghost.index.values)


G_data = nx.from_pandas_edgelist(fb_df_partial, "node_1", "node_2", create_using=nx.Graph())
```

شکل ۰-۲۱: استخراج ویژگی ها از جدول link

نتیجه‌گیری

۴-۱- خروجی کدها

همان‌طور که در فصل سه نشان داده شد گراف خروجی که از داده‌های اینترنت اجتماعی اشیا به دست آمد به شکل زیر است.



شکل ۴-۱ : گراف ایجاد شده از داده‌های اینترنت اجتماعی
اشیا

حال مدل node2vec را به بر روی گرافی که داریم آموزش می‌دهیم.

```
node2vec = Node2Vec(G_data, dimensions=100, walk_length=16, num_walks=50)

n2w_model = node2vec.fit(window=7, min_count=1)
```

شکل ۴-۲ : آموزش مدل node2vec به گراف اینترنت
اجتماعی اشیا

مدل آموزش داده شـــده node2vec را بر روی هر جفت گره موجود در جدول data که ایجاد کرده ایم اجرا می‌کنیم.برای محاسبه ویژگی‌های یک جفت گره یا یک یال، ویژگی گره‌ها را در آن جفت گره جمع می‌کنیم.

```
x = [(n2w_model[str(i)]+n2w_model[str(j)]) for i,j in zip(data['node_1'], data['node_2'])]

xtrain, xtest, ytrain, ytest = train_test_split(np.array(x), data['link'],
                                                test_size = 0.3,
                                                random_state = 35)
```

شکل ۴-۳ : اضافه کردن ویژگی‌ها در یک جفت گره

برای بررســـی عملکرد این مدل از امتیاز AUC-ROC اســـتفاده می‌کنیم. برای محاســـبه این امتیاز از مدل regression استفاده می‌کنیم.

```
parameters = {
    'objective': 'binary',
    'metric': 'auc',
    'is_unbalance': 'true',
    'feature_fraction': 0.5,
    'bagging_fraction': 0.5,
    'bagging_freq': 20,
    'num_threads' : 2,
    'seed' : 76
}

model = lgbm.train(parameters,
                   train_data,
                   valid_sets=test_data,
                   num_boost_round=1000,
                   early_stopping_rounds=20)
```

شکل ۴-۴ : بررسی عملکرد مدل node2vec با استفاده از مدلregression

آموزش پس از ۲۰۸ بار تکرار متوقف شـــد زیرا معیارهای توقف اولیه را اعمال کردیم. و در نهایت این مدل امتیاز قابل توجه AUC 0.9273 را در آزمایش دریافت کرد.

مراجع

1. https://link.springer.com/chapter/10.1007/978-3-030-49435-3_7
2. Bleecker, J.: A manifesto for networked objects - cohabiting with pigeons, arphids and aibos in the Internet of Things. In: Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI), pp. 1–17 (2006)
3. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.-W.: Smart-its friends: a technique for users to easily establish connections between smart artefacts. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 116–122. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45427-6_10
4. https://www.ericsson.com/en/blog/2012/4/a-social-web-of-things**.**
5. Atzori, L., Iera, A., Morabito, G., Nitti, M.: The Social Internet of Things (SIoT) - when social networks meet the Internet of Things: concept, architecture and network characterization. Comput. Netw. **56**(16), 3594–3608 (2012)
6. Nitti, M., Atzori, L., Cvijikj, I.P.: Friendship selection in the social internet of things: challenges and possible strategies. IEEE Internet Things J. **2**(3), 240–247 (2015)
7. Roopa, M., Pattar, S., Buyya, R., Venugopal, K.R., Iyengar, S., Patnaik, L.: Social Internet of Things (SIoT): foundations, thrust areas, systematic review and future directions. Comput. Commun. **139**(1), 32–57 (2019)
8. Atzori, L., Iera, A., Morabito, G.: From smart objects to social objects: the next evolutionary step of the internet of things. IEEE Commun. Mag. **52**(1), 97–105 (2014)
9. Tran, N.K., Sheng, Q.Z., Babar, M.A., Yao, L.: Searching the web of things: state of the art, challenges, and solutions. ACM Comput. Surv. (CSUR) **50**(4), 1–34 (2017)
10. Tran, N.K., Sheng, Q.Z., Babar, M.A., Yao, L., Zhang, W.E., Dustdar, S.: Internet of Things search engine. Commun. ACM **62**(7), 66–73 (2019)
11. Girau, R., Martis, S., Atzori, L.: Lysis: a platform for IoT distributed applications over socially connected objects. IEEE Internet Things J. **4**(1), 40–51 (2017)

12. Ortiz, A.M., Hussein, D., Park, S., Han, S.N., Crespi, N.: The cluster between internet of things and social networks: review and research challenges. IEEE Internet Things J. **1**(3), 206–215 (2014)
13. A. Khelloufi., H. Ning., S. Dhelim.,T.Qiu., J. Ma., R.Huang, and L.Atzori, "A social Relationships Based Service Recommendation System For Siot Devices" vol. 14, no. 8, august 2015
14. G. Omale, "Gartner identifies top 10 strategic iot technologies and trends," Gartner website, 2018
15. A. P. Fiske, "The four elementary forms of sociality: Framework for a unified theory of social relations," Psychological Review, vol. 99, no. 4, pp. 689–723, 1992.
16. T. Zhu, S. Dhelim, Z. Zhou, S. Yang, and H. Ning, "An architecture for aggregating information from distributed data nodes for industrial internet of things," Computers & Electrical Engineering, vol. 58, pp. 337–349, Feb. 2017. [Online]. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2016.08.018 [8] J.
17. A. Grover and J. Leskovec , Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data MiningAugust 2016 Pages 855–864
18. G. Tsoumakas and I. Katakis. Multi-label classification: An overview. Dept. of Informatics, Aristotle University of Thessaloniki, Greece, 2006.
19. http://www.social-iot.org/index.php?p=downloads

Shahid Rajaee Teacher Training University
Faculty of Computer Engineering
Department of software
B. Sc. Thesis

Title:

Predictive system in the social Internet of Things

Supervisor:
Dr.Sahar kianian

By:
Zahra Valadi

Winter 2023