

گزارش ارزیابی کارایی سیستم‌های اینترنت اشیا پزشکی و اینترنت اشیا براساس مجموعه داده‌های CICIoMT2024 و مدلینگ ریاضیاتی استاد ناظر: آقای دکتر مهدی امینیان

علیرضا سلطانی نشان

۲۵ دی ۱۴۰۳

مجوز

به فایل license همراه این برگه توجه کنید. این برگه تحت مجوز GPLV۳ منتشر شده است که اجازه نشر و استفاده (کد و خروجی/pdf) را رایگان می‌دهد.

فهرست مطالب

۳	۱ رابطه مدل‌های ریاضی با شاخص‌های کلیدی کارایی
۳	۱.۱ KPI ها
۳	۲.۱ مدل ها
۹	۲ شاخص‌های محاسبه و ارزیابی عملکرد
۹	۱.۲ فرمول شانون
۹	۲.۲ فرمول محاسبه بار سیستم یا System load
۱۰	۳.۲ تاخیر سرویس‌دهی یا Service latency
۱۰	۱.۳.۲ بخش‌هایی که زمان سرویس‌دهی دارند
۱۱	۲.۳.۲ زمان ارتباطی
۱۱	۳.۳.۲ زمان پردازشی
۱۲	۴.۳.۲ زمان پردازش محلی t_L یا زمان پردازش در هر زیر سیستم t_{pi}
۱۲	۵.۳.۲ تابع محاسبه CPU time
۱۳	۶.۳.۲ زمان پردازش محلی با توجه به اندازه داده (D)
۱۳	۴.۲ مصرف انرژی
۱۳	۱.۴.۲ مجموع مصرف انرژی E_{dev}
۱۵	۳ مدل‌های ارزیابی کارایی
۱۵	۱.۳ ارتباط بین عملکرد و ویژگی‌های زیرساخت IoT
۱۵	۲.۳ مشکل مقایسه KPIs مختلف
۱۶	۳.۳ استفاده از تابع سودمندی Utility function
۱۷	۴.۳ مدل‌سازی توابع سودمندی برای هر KPI

۵.۳	فرمول تابع سیگموئید	۱۸
۶.۳	اهداف فرمول‌های سیگموئید	۱۸

فهرست تصاویر

۴	دسته‌بندی دامنه استفاده از سیستم‌های IoT	۱
۵	حوزه‌های تخصصی بخش اپلیکیشن در IoT	۲
۶	حوزه‌های بخش میان‌افزار در IoT	۳
۷	حوزه‌های بخش شبکه در IoT	۴
۸	حوزه‌های بخش Embedded در IoT	۵

فهرست جداول

۹	تعریف ثابت‌های مورد استفاده در فرمول‌ها	۱
---	---	---

مهم‌ترین انگیزه برای توسعه این تحقیق وجود کمبود در داده‌های موجود ارزیابی کارایی تجهیزات اینترنت اشیا پزشکی و پیشرفت امنیتی تمام شبکه‌هایی که در خصوص جریان‌های داده‌ای و پردازش داده‌های پزشکی کار می‌کنند، می‌باشد بخصوص برای دستگاه‌های اینترنت اشیا پزشکی به دلیل اطلاعات حیاتی‌ای که می‌توان به واسطه آن‌ها از بیماران با بیماری‌های مختلف مانیتور و دریافت کرد.

۱ رابطه مدل‌های ریاضی با شاخص‌های کلیدی کارایی

برای مدل‌سازی جهت ارزیابی عملکرد سیستم‌های مختلف از قبیل سیستم‌های IoT بایستی شاخص‌های کلیدی کارایی یا KPI^۱ را بشناسیم. این KPIها شامل مجموعه‌ای از فرمول‌های ریاضیاتی هستند که نسبت به نیازمندی‌های غیرعملیاتی یا Non-functional requirements متفاوت می‌باشند. نیازمندی‌های غیرعملیاتی مانند میزان مصرف انرژی، میزان مصرف حافظه، سرعت انتقال اطلاعات و گزردهی، میزان دقت، میزان در دسترس بودن، میزان قابلیت اطمینان و غیره می‌باشد. برای هر کدام از این نیازمندی‌های غیرعملیاتی فرمول‌هایی وجود دارد که می‌تواند با فراگیری آن‌ها به مقادیری رسید که در مرحله بعد می‌توان آن‌ها را در مدل‌های ریاضیاتی استفاده کرد. مرحله دوم در حقیقت ارزیابی کارایی سیستم‌ها با استفاده از مدل‌های شناخته شده ریاضیاتی می‌باشد که رابطه مستقیمی با KPIها دارند. یعنی نمی‌توان مقادیر KPI را برای این مدل‌ها نادیده گرفت. در نهایت با مدل‌های مطرح شده می‌توان کارایی سیستم‌های IoT در دامنه‌های مختلف را ارزیابی کرد و بین آن‌ها مقایسه و Trade off نسبت به Benchmarkهای بدست آمده انجام داد.

KPIها و مدل‌هایی که در این گزارش مورد بررسی قرار گرفته‌اند:

۱.۱ KPIها

•

۲.۱ مدل‌ها

•

یکپارچه‌سازی سیستم‌های IoT با سیستم‌های ابری چالش‌های زیادی داشته مثل:

- تاخیرهای شبکه‌ای
- گزردهی
- مصرف انرژی
- قابلیت اطمینان

یه سری مفاهیم جدیدی در حوزه پردازش‌ها مطرح شده که حتی می‌تواند کاربردهای مختلفی در استفاده از اینترنت اشیا باشد. این مفاهیم جدید مثل Fog computing، edge computing، mobile edge computing، mobile cloud computing و Cloudlet ها هستند. در این مقاله یک مدل ریاضیاتی برای توصیف رسمی سیستم‌های IoT ارائه داده شده است. علاوه بر این یک ارزیابی آنالیز شده برای طراحی این سیستم‌ها با استفاده از مطابقت با معماری، تکنولوژی‌ها، پروتکل‌ها و مدل‌های یکپارچه‌سازی برای بهینه‌سازی عملکرد نیز ارائه می‌دهد.

Approach of this article:

بعد از خواندن این مقاله به یک روش بهینه برای بهینه‌سازی کارایی مبتنی بر فرایندهای offloading مانند load balancing آشنا می‌شیم. مدلینگ ریاضیاتی سیستم‌های IoT یک نمایی از سیستم ایجاد می‌کنند که به فهمیدن المان‌ها، تعاملاتشون، و رفتارهاشون کمک می‌کند.

۱. مدل مفهومی یا conceptual model یک ساختار سطح بالایی برای توصیف عملیاتی است که در سیستم‌های IoT انجام می‌شود.

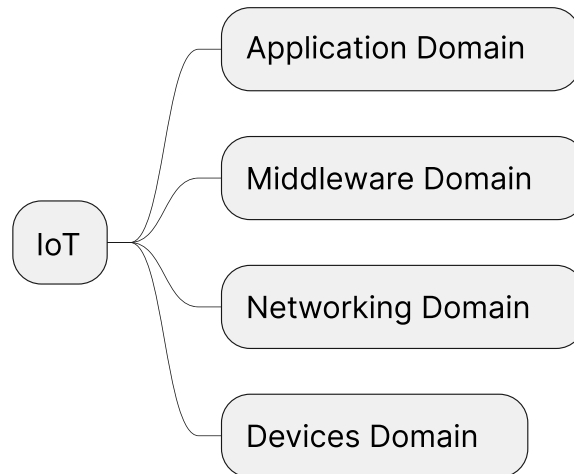
۲. مدل رفتاری یا behavioral model ممکنه شامل جزئیات باشد. مثل جریان داده بین المان‌ها.

به طور کلی مدلینگ به مشخص شدن و پاسخ به مسائل مربوط به کارایی کمک بسزایی می‌کنه و اجازه میده که سیستم‌ها بهینه‌تر، کارا تر و مطمئن‌تر باشن.

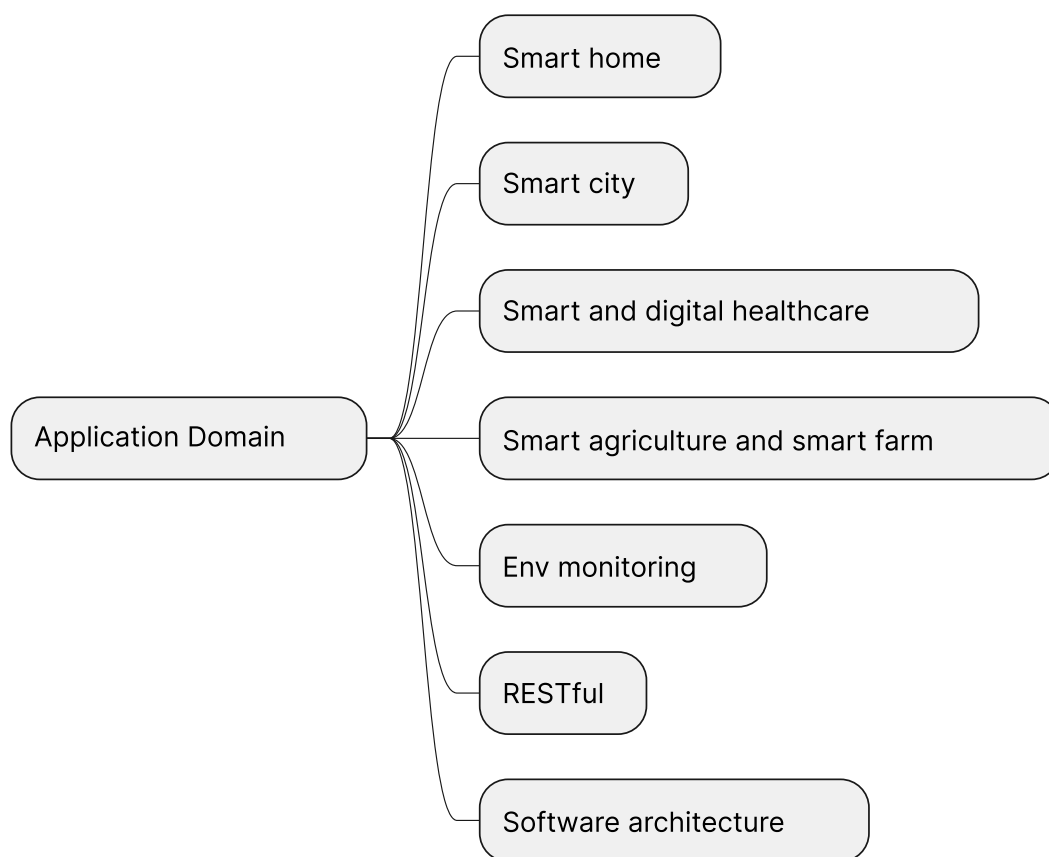
هر موقع در مورد مدلینگ یک سیستم IoT صحبت میشه در حقیقت قراره یه چهارچوبی درست بشه که بتونیم باهاش تست کنیم، تایید یا validation انجام بدیم و یا بتوانیم سیستم را به تقاضاهایی که داریم optimize کنیم.

^۱ Key Performance Indicators

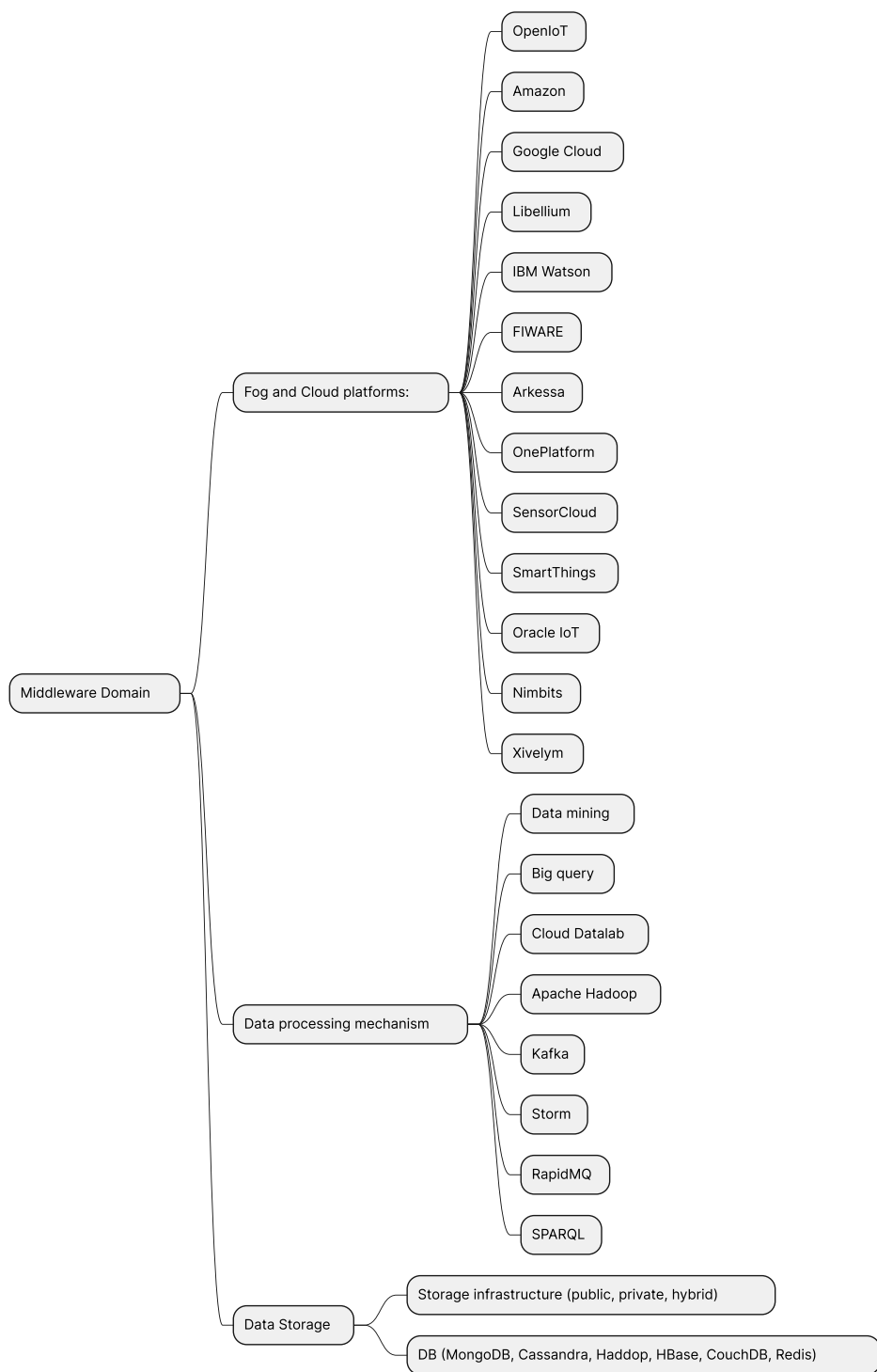
فرایند سیستم‌های IoT معمولاً شامل شناسایی، دریافت اطلاعات، (Sensing) فعالیت‌های تحت شبکه، و محاسبات کوچک هستند که باعث می‌شود همیشه با محیط فیزیکی و هر اشیایی ارتباط برقرار کند. در این مقاله سیستم‌های IoT با کاربردی که دارند به ۴ دسته تقسیم می‌شوند:



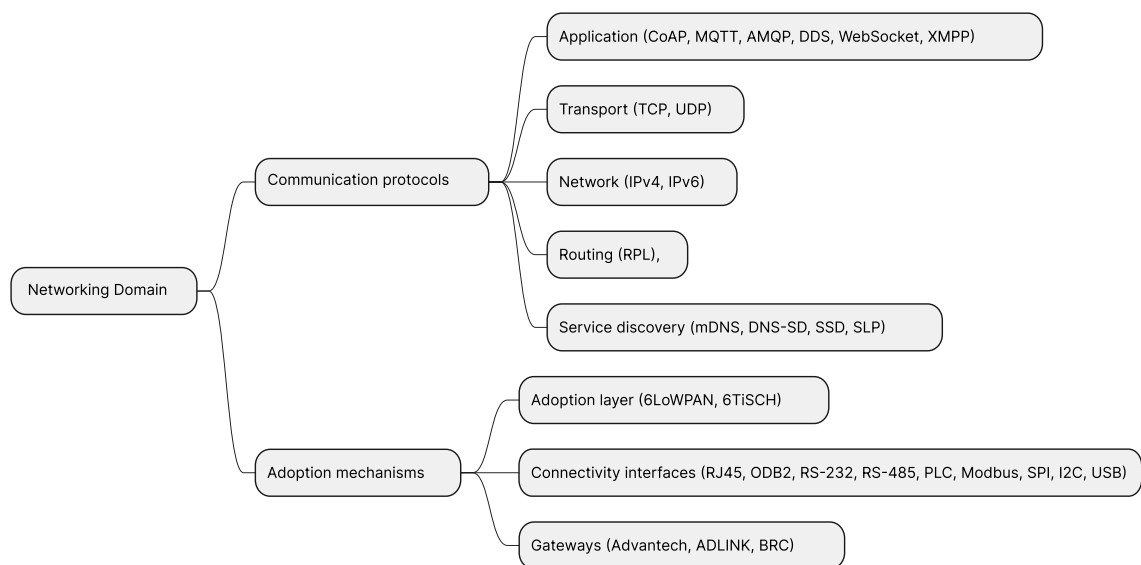
شکل ۱: ۴ دسته‌بندی دامنه استفاده از سیستم‌های IoT



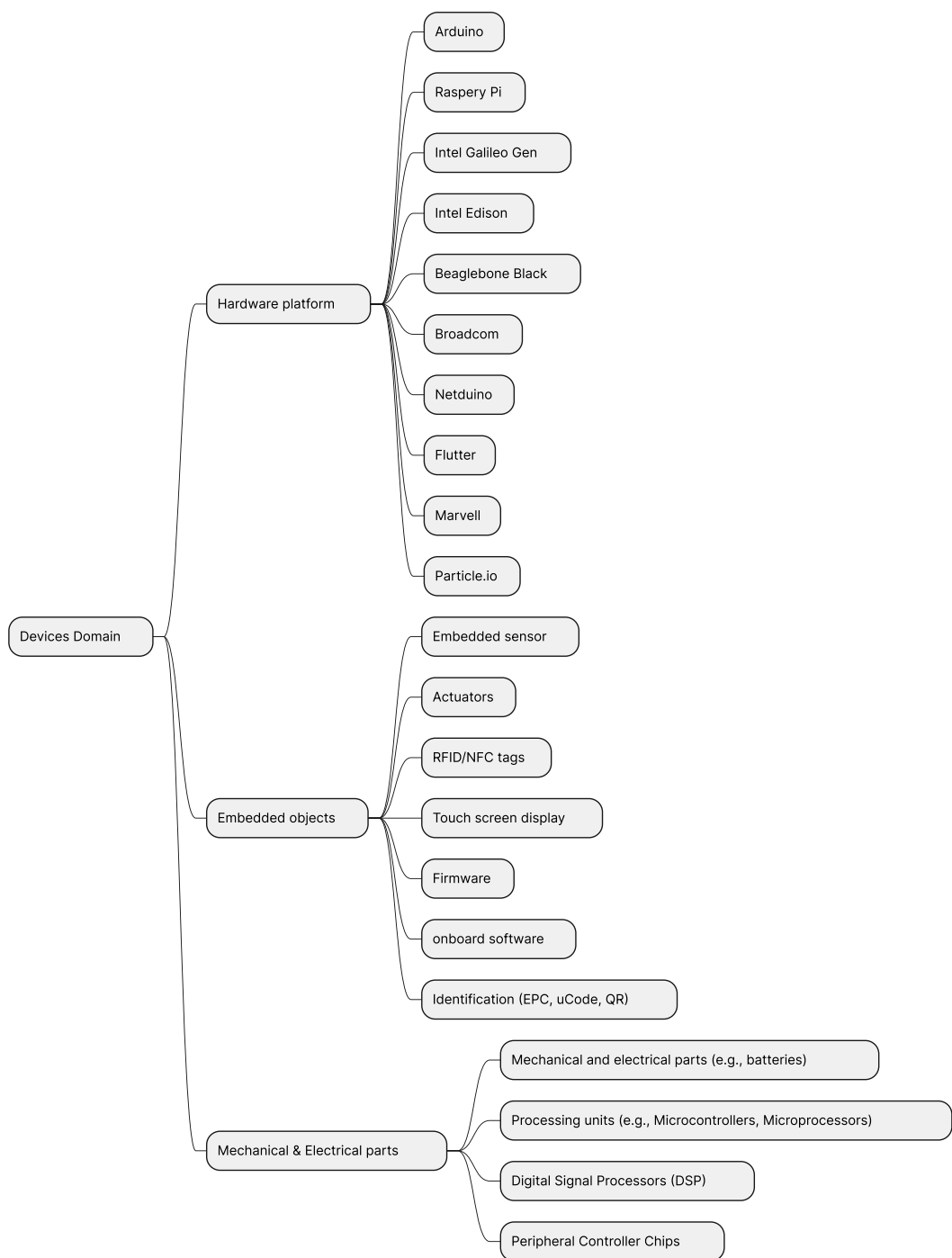
شکل ۲: حوزه‌های تخصصی بخش اپلیکیشن در IoT



شکل ۳: حوزه‌های بخش میان‌افزار در IoT



شکل ۴: حوزه‌های بخش شبکه در IoT



شکل ۵: حوزه‌های بخش Embedded در IoT

عمومیت جریان داده در سیستم‌های IoT:

- دریافت داده
- انتقال داده
- پردازش داده

- ذخیره‌سازی داده
- آنالیز و معنادار کردن داده

۲ شاخش‌های محاسبه و ارزیابی عملکرد

تعاریف	ثابت‌ها
نرخ ورود اطلاعات	D_{rate}
مصرف انرژی	E_{dev}
تاخیر سرویس‌دهی	$T_{exe.}$
مشخصات سیستم IoT	IoT_{sysp}
زمان ورکلودها	t_{ws}
تعداد دستگاه‌ها	k
درصد داده‌ای که در سیستم i پردازش می‌شود	β
مقدار گذردهی	τ
نیازمندی مشخص کارایی P_{ireq} جایی که P_i مقدار از کارایی است	$IoT_{sys}(P) = \{P_{ireq}, P_i\}$
مدت زمان سرویس‌دهی	$P_i = \langle T_{exe.}, E_{dev} \rangle$
مصرف انرژی داخلی	E_{loc}
انرژی مورد نیاز برای offloading	E_{off}
انرژی مصرفی در زمان بیکاری	E_{idle}
پردازش تسک‌های داخل دستگاه	P_L
زمان پردازش تسک‌های داخلی	t_L

جدول ۱: تعریف ثابت‌های مورد استفاده در فرمول‌ها

۱.۲ فرمول شانون

نرخ داده‌ها می‌تواند از طریق فرمول شانون محاسبه شود.

$$D_{rate} = B_{i,j} \log_2 \left(1 + \frac{|h_{ij}|^2 \cdot P_{tx}}{P_{Nj}} \right) \quad (1)$$

- B_{ij} : پهنای باند
- h_{ij} : بهره کانال بین دستگاه مبدا و مقصد که نشان می‌دهد سیگنال چگونه در مسیر بین فرستنده و گیرنده تقویت یا تضعیف می‌شود.
- P_{tx} : توان ارسال
- P_N : میزان نویز مقصد

کاربرد زیادی در سناریوهایی دارد که در آن یک ارسال کننده و یک دریافت کننده وجود دارد.

۲.۲ فرمول محاسبه بار سیستم یا System load

مجموع بار سیستم از طریق فرمول زیر بدست می‌آید:

$$D = \sum_{k=1}^N D_{rate,k} \times T_{w,k} \quad (2)$$

- $D_{rate,k}$: نرخ تولید داده توسط دستگاه IoT

- $t_{w,k}$: مدت زمان ورودی در دستگاه IoT یا به عبارتی دیگر، مدت زمانی که طول می‌کشد یک دستگاه IoT ورودی را دریافت و سپس آن را پردازش و هندل کند.

۳.۲ تاخیر سرویس‌دهی یا Service latency

مسئله service latency یا ($T_{exe.}$) service execution time مدت زمانی است که طول می‌کشد سیستم IoT تمام درخواست‌های پردازشی و ارتباطی را اجرا کند (the total application exe time). مدت زمان کل مصرف شده از، مدت زمان سرویس یک ریکوئست به مدت زمان تمام تسک‌هایی که با موفقیت پردازش شده‌اند. بنابراین، این فاصله زمانی بین درخواست برنامه و بدست آوردن نتایج می‌باشد.

۱.۳.۲ بخش‌هایی که زمان سرویس‌دهی دارند

- مدت زمان انتقال داده از دستگاه IoT به زیر ساخت فاگ
- مدت زمان انتقال داده از دستگاه IoT به سرورهای ابری
- مدت زمان انتقال داده از فاگ به کلاد
- مدت زمان انتقال اعلانات از کلاد به فاگ
- مدت زمان انتقال اعلانات از کلاد به دستگاه‌های IoT
- مدت زمان انتقال اعلانات از فاگ به دستگاه IoT
- مدت زمان محاسبات در دستگاه IoT
- مدت زمان محاسبات در لایه فاگ
- مدت زمان محاسبات در سرورهای ابری

نکته: مدت زمانی که برای هر کاری در سیستم‌های IoT سپری می‌شود به نوع و شیوه پیاده‌سازی معماری دستگاه‌ها و نرم‌افزار بخش‌ها بستگی دارد و می‌تواند کاملاً متفاوت باشند. عموماً سرویس لیتنسی بین المان‌های سیستم IoT توزیع شده هستش و شامل دستگاه‌های اینترنت اشیا، شبکه‌ها و سیستم‌های پردازشی می‌شود.

تاخیر سرویس‌دهی:

$$T_{exe.} = T_{cm} + T_{cp} \quad (۳)$$

- T_{cm} : مدت زمان تاخیر در ارتباطات

- T_{cp} : مدت زمان تاخیر در پردازش

مدت زمان اجرا بایستی کمتر از زمان‌بندی تسک‌ها در فاگ یا کلاد باشد. یعنی سرویس تایم باید کمتر از نیازمندی‌های IoT application (T_{req}) باشد.

برای کاهش service latency از فرمول زیر بایستی پیروی کند:

$$Objective : \min(T_{exe.}) = T_{cm} + T_{cp} \leq T_{req} \quad (۴)$$

موقعی که داری در مورد Execution time می‌نویسی فرمول communication latency رو باید داشته باشی که بگی از کجا بدست میاد. و همچنین فرمول computation latency رو هم بدش. یعنی ثابت‌ها خودشون از زیر ثابت‌های اصلی بدست میان که جمع میشن و میشه T_{exe} .

۲.۳.۲ زمان ارتباطی

$$T_{cm} = \sum_{i=1}^N (d_{proc} + d_{queue} + d_{trans} + d_{prop}) \quad (۵)$$

• d_{prop} : تاخیر پردازشی

• d_{queue} : تاخیر در صف

• d_{trans} : تاخیر انتقال

• d_{prop} : تاخیر توزیع

تاخیر مربوط به Propagation مجموع زمان مورد نیاز برای داده جهت ارسال از منبع به مقصد که مبتنی بر طول لینک فیزیکی و سرعت رسانی می باشد.

$$d_{trans} = \frac{P_s}{R_L} \quad (۶)$$

که در آن:

• P_s : اندازه بسته در واحد bits

• R_L : سرعت لینک ارتباطی bps

$$d_{prop} = \frac{l_{ij}}{c} \quad (۷)$$

• l_{ij} : لینک فیزیکی

• c : سرعت توزیع media

۳.۳.۲ زمان پردازشی

$$T_{cp} = T_L + \sum_{i=1}^k t_{offi} \quad (۸)$$

که در آن:

• t_L : اجرا و پردازش های داخلی

• t_{offi} : اجرا و پردازش های خارج از دستگاه IoT مانند برنامه هایی که در سیستم های ابری یا Fog مستقر شده اند که وظیفه پردازش تسک های Offloading را دارند.

به بیان دیگر می توان آن را به صورت مدل زیر محاسبه کرد:

$$T_{cp} = t_L + t_F + t_C \quad (۹)$$

$$T_{cp} = t_L + \max_{i=1, \dots, k} t_{Fi} + \max_{j=1, \dots, n} t_{Cj} \quad (۱۰)$$

که در آن:

- t_L : مدت زمان پردازش‌های داخلی

- t_{Fi} : مدت زمان پردازش در نود i^{th} در Fog

- t_{Ci} : مدت زمان پردازش در سرور i^{th} ابری

عموماً مصرف پردازشی بستگی به سرعت و معماری پردازنده مرکزی (CPU)، حافظه رم (RAM)، سرعت حافظه ذخیره‌ساز (HDD) یا (SSD)، سرعت پردازنده گرافیکی یا (GPU) و غیره دارد.

۴.۳.۲ زمان پردازش محلی t_L یا زمان پردازش در هر زیر سیستم t_{pi}

برای بدست آوردن زمان پردازش در هر زیر سیستم از فرمول زیر استفاده می‌شود:

$$t_{pi} = \frac{ICC_i}{f_{cpu,i}} \quad (11)$$

- t_{pi} : زمان پردازشی در زیر سیستم i

- ICC_i : تعداد سایکل‌های CPU که برای اجرای یک برنامه نیاز است.

- $f_{cpu,i}$: نرخ کلاک (فرکانس کاری CPU) زیر سیستم i

$$t_{Pi} = t_{CPU_i} + t_{I/O_i} \quad (12)$$

۵.۳.۲ تابع محاسبه CPU time

مدت زمانی که در CPU برای اجرا برنامه در نظر گرفته می‌شود به دو دسته تقسیم می‌شود:

۱. User CPU time

۲. System CPU time: t_{OS}

محاسباتی که در CPU time انجام می‌شود خالصانه در قسمت پردازشگر مرکزی صورت می‌گیرد و هیچ محاسبه جانبی مانند مدت زمان I/O و مدت زمان اجرای دیگر برنامه‌ها در نظر گرفته نمی‌شود.

$$t_{cpu_i} = \frac{ICC_i}{f_{cpu_i}} + t_{OS} = ICC_i \times t_{cc_i} + t_{OS} \quad (13)$$

حاصل این تابع معمولاً بسیار کوچک است و می‌تواند نادیده گرفته شود زیرا به سمت صفر میل می‌کند ($t_{OS} \rightarrow 0$). به همین خاطر بیشتر روی User CPU time تمرکز می‌کند که توسعه‌دهنده بر روی آن کدهای خود را اجرا می‌کند و سیستم IoT را راه‌اندازی می‌کند.

تابع مطرح شده بر اساس قدرت محاسباتی دستگاه (f_{cpu_i}) و تعداد کلاک CPU برای اجرای یک برنامه (ICC_i) می‌باشد. مقدار f_{cpu_i} بر واحد (Hz) می‌باشد و t_{cc_i} زمان چرخه کلاک است. لازم به ذکر است که ICC_i به نوع دستورالعمل که شامل اندازه داده ورودی، زبان برنامه نویسی، میزان پیچیدگی الگوریتم نرم‌افزاری مورد استفاده، و دیگر موارد می‌باشد.

بخش CPI (Clock cycles per instruction) به عنوان میانگین تعداد چرخه کلاک است که هر دستورالعمل به آن نیاز دارد. اگر I_{app_j} تعداد دستورالعمل‌ها برای یک برنامه باشد آن وقت ICC_i از طریق معادله زیر بدست می‌آید.

$$ICC_i = \sum_{j=1}^k I_{app_j} \times CPI_j \quad (14)$$

۶.۳.۲ زمان پردازش محلی با توجه به اندازه داده (D)

$$I_{CC_i} = X \times D \quad (15)$$

- D : اندازه ورودی داده بر حسب بیت
 - X : شدت پردازش (تعداد چرخه‌های مورد نیاز برای هر بیت داده)
- بنابراین خواهیم داشت:

$$t_{pi} = \frac{\beta_i \times D \times X}{f_{cpu,i}} \quad (16)$$

که در آن β_i درصد داده‌ای است که در سیستم i پردازش می‌شود.

۴.۲ مصرف انرژی

۱.۴.۲ مجموع مصرف انرژی E_{dev}

برای محاسبه مصرف کل انرژی در سیستم‌های IoT می‌توان از فرمول زیر استفاده کرد:

$$E_{dev} = F(E_{cp}, E_{cm}, E_{idle}, E_{other}) \quad (17)$$

- E_{cp} : انرژی مصرف شده طی محاسبات
 - E_{cm} : انرژی مصرف شده در طی ارتباطات
 - E_{idle} : انرژی مصرف شده در حالت نرمال و بیکار سیستم
 - E_{other} : انرژی مصرف شده توسط بقیه فرایندها مانند سنسورها، صفحه نمایش، کارت گرافیک و غیره.
- یک مدل برای بررسی مصرف انرژی توسط دستگاه‌های IoT که شامل فرایندهای پردازشی و ارتباطی می‌شود عبارت است از:

$$E_{dev} = E_{loc.} + E_{off.} \quad (18)$$

میزان انرژی مورد نیاز برای پردازش داخلی به تسک‌های داخلی وابسته می‌باشد:

$$E_{loc.} = P_L \times t_L \quad (19)$$

محاسبه میزان انرژی پردازشی از حاصل ضرب I_{CC_i} و انرژی مصرفی CPU به ازای هر چرخه CPU بدست می‌آید:

$$E_{loc.} = k \times I_{CC_i} \times f_{cpu_i}^2 = k \times D \times X \times f_{cpu_i}^2 \quad (20)$$

- k : ثابتی است که به مشخصات سخت‌افزار مربوط است.
- D : اندازه داده ورودی بر مبنای bits

• X : شدت یا داده ورودی محاسباتی

محاسبه انرژی برای انجام تسک‌های offloading نیازمند انرژی برای ارسال داده‌ها و دریافت نتایج آن می‌باشد که با E_{comm} نمایش می‌دهند. لازم به ذکر است مدت زمانی که طول می‌کشد سیستم نتایج را دریافت کند سیستم در وضعیت idle باقی مانده است. به همین صورت برای بدست آوردن مصرف انرژی برای تسک‌های offloading از فرمول زیر استفاده می‌شود:

$$E_{off} = E_{cm} + E_{idle} \quad (21)$$

در واقع E_{id} انرژی مورد استفاده دستگاه IoT در زمانی که دستگاه در حالت بیکار می‌باشد. این بیکاری به منظور آن است که دستگاه IoT در حال انتظار برای دریافت نتیجه از سرورها می‌باشد.

$$E_{idle} = P_{idle} \times t_{off} \quad (22)$$

$$t_{off} = T_{exe} - (t_L + T_{cm}) \quad (23)$$

مجموع انرژی مصرفی جهت انتقال داده‌ها با انرژی مصرفی در هنگام دریافت داده‌ها ما را به انرژی مصرفی ارتباطی می‌رساند:

$$E_{cm} = E_{tx} + E_{rx} \quad (24)$$

در یکی از کارها [۹] مدلی برای مصرف انرژی نسبت به انتقال و جا به جایی داده‌ها مطرح شده که سطوح مختلف مصرف باتری را برای uplink و downlink شامل می‌شود:

$$P_{tx} = p_u \tau_u + \beta \quad (25)$$

$$P_{rx} = p_d \tau_d + \beta \quad (26)$$

گذردهی uplink، τ_u می‌باشد و گذردهی downlink، τ_d است. در حالی که p_u و p_d میزان انرژی مورد نیاز برای انتقال داده‌ها در uplink و downlink می‌باشد. ثابت β میزان مصرف انرژی در حالت idle می‌باشد. این مقادیر کاملاً به تکنولوژی ارتباطی، پروتکل‌ها و دستگاه‌هایی که روی آن برنامه مستقر شده است وابسته می‌باشد. برای انتقال همزمان uplink و downlink سطح انرژی می‌تواند با فرمول زیر محاسبه شود:

$$P_{trx} = p_u \tau_u + p_d \tau_d + \beta \quad (27)$$

نسبت معادله uplink بر روی downlink به همراه پهنای باند، می‌تواند فرمول را برای محاسبه انرژی بهینه شبکه برای انتقال یک مقدار داده معین (energy per bit). مقدار انرژی مورد نیاز برای ارسال از طریق فرمول $E(D)_{rx}$ میزان انرژی مورد نیاز برای دریافت داده‌ها از طریق فرمول $E(D)_{tx}$ حاصل می‌شود.

$$E(D)_{tx} = p_u + \beta \tau_u^{-1} \quad (28)$$

$$E(D)_{rx} = p_d + \beta \tau_d^{-1} \quad (29)$$

مقدار داده‌هایی که بر واحد بیت توسط دستگاه‌های IoT ارسال و دریافت می‌شوند به ترتیب D_{tx} و D_{rx} می‌باشند. با در نظر گرفتن محاسبات پیشین، می‌توان در نهایت میزان مصرف انرژی توسط دستگاه‌های IoT را به شکل زیر بدست آورد:

$$E_{dev} = (P_L \times t_L) + (P_{tx} \times t_{tx}) + (P_{rx} \times t_{rx}) + (P_{id} \times t_{off}) \quad (30)$$

$$E_{dev} = (P_L \times t_L) + ((p_u \tau_u + \beta) \times t_{tx}) + ((p_d \tau_d + \beta) \times t_{rx}) + (P_{id} \times t_{off}) \quad (31)$$

$$E_{dev} = (P_L \times t_L) + ((p_u + \beta \tau_u^{-1}) \times D_{tx}) + ((p_d + \beta \tau_d^{-1}) \times D_{rx}) + (P_{id} \times t_{off}) \quad (32)$$

یکی از مهم‌ترین چالش‌های دستگاه‌های IoT مربوط به مصرف باتری آن‌ها می‌باشد. در بعضی مواقع دستگاه‌های IoT از باتری‌هایی استفاده می‌کنند که شرایط جایگزین کردن آن‌ها وجود ندارد. هر دستگاه IoT حتی در حالت بیکار انرژی بابت، پردازش داده‌ها، ارسال و دریافت داده‌ها مصرف می‌کنند. انرژی موجود $E_{dev}(t)$ در طی زمان کاهش پیدا می‌کند. به همین ترتیب انرژی باقی‌مانده $E_{dev}(r)$ یا مدت زمانی که سیستم می‌تواند روشن بماند از طریق فرمول زیر بدست می‌آید.

$$E_{dev}(r) = E_{dev}(i) - E_{dev}(t) \quad (33)$$

• $E_{dev}(i)$: مقدار اولیه انرژی دستگاه

مقدار باتری باقی‌مانده $T(sys)$ به میزان ظرفیت باطری یا انرژی باقی‌مانده و انرژی مورد نیاز دستگاه برای انجام تمام سرویس‌های دستگاه، بستگی دارد. انرژی مصرفی وابسته به قدرت مورد نیاز برای پردازش‌های داخلی P_{cp} انتقال داده‌ها P_{cm} و بقیه فرایندها P_{other} می‌باشد.

$$T(sys) = \frac{E_{dev}(r)}{P_{cp} + P_{cm} + P_{other}} \quad (34)$$

یکی دیگر از چالش‌های دستگاه‌های IoT مربوط به منبع‌تغذیه آن‌ها می‌باشد. در مواقعی که دستگاه‌های IoT از باتری استفاده می‌کنند و به دلیل محیط‌های مختلف شرایط به گونه‌ای است که امکان تعویض باتری وجود ندارد، محدودیت‌های باتری اغلب به عنوان شاخص طول عمر دستگاه‌های IoT استفاده می‌شود و می‌تواند به عنوان یکی از مهم‌ترین معیارهای QoS مورد استفاده قرار گیرد. هدف اصلی در این دستگاه‌ها این است که مصرف انرژی به حداقل برسد و طول عمر کلی سیستم به بیشترین حد ممکن.

۳ مدل‌های ارزیابی کارایی

در این بخش بررسی می‌شود که چگونه می‌توان عملکرد سیستم‌های IoT را ارزیابی کرد وقتی چندین شاخص کلیدی عملکرد KPIs با واحدها و اهمیت‌های مختلف وجود دارد.

۱.۳ ارتباط بین عملکرد و ویژگی‌های زیرساخت IoT

عموماً عملکرد یک برنامه به ویژگی‌های زیرساخت IoT مانند توان محاسباتی، تاخیر شبکه، مصرف انرژی، و غیره وابسته است. هر تغییری در زیرساخت IoT می‌تواند تاثیر مستقیمی بر عملکرد برنامه داشته باشد.

۲.۳ مشکل مقایسه KPIs مختلف

هر KPI مانند تاخیر شبکه، گذردهی، پهنای باند، مصرف انرژی واحد و مقیاس خاص خود را دارد. برای مثال تاخیر بر حسب میلی‌ثانیه ms اندازه‌گیری می‌شود و گذردهی بر حسب mips و مصرف انرژی بر حسب وات-ساعت. این تفاوت در واحدها باعث می‌شود مقایسه مستقیم آن‌ها دشوار شود. علاوه بر مختلف بودن واحدها و معیارها، برخی از KPIها ممکن است از اهمیت بیشتری برخوردار باشند که معمولاً به هر شاخص وزنی یا weight اختصاص می‌یابد.

۳.۳ استفاده از تابع سودمندی Utility function

در اینجا یک تابع کاربردی به نام $IoT_{sys}(p)$ وجود دارد که عملکرد سیستم IoT را براساس KPIهای نرمال شده و وزن دهی شده ارزیابی می کند. این تابع می تواند برای مقایسه مدل های مختلف سیستم IoT استفاده شود و مشخص کند که کدام مدل بهتر است.

$$IoT_{sys}(p) = \sum_{i=1}^n w_i \times f(p_i) \quad (35)$$

- n : تعداد شاخص های کلیدی کارایی
 - (p_i) : ارزیابی کارایی سیستم های IoT
 - $f(p_i)$: مطابقت با تابع utility مشخص برای ارزیابی هر KPI که بین دو عدد ۰ و ۱ تبدیل (نرمال سازی) شده است.
 - w_i : وزن ضرایب برای تعیین آن که کدام KPI در سیستم IoT مورد نظر مهم تر می باشد.
- به عنوان مثال، یک سیستم IoT داریم که برای پایش سلامتی بیماران طراحی شده است. می خواهیم عملکرد این سیستم را ارزیابی کنیم. در اینجا چند KPI مهم وجود دارد:

۱. زمان تاخیر یا Latency: مدت زمانی که طول می کشد داده ها از سنسور به سیستم مرکزی برسند. (بر حسب میلی ثانیه)
۲. مصرف انرژی یا Energy consumption: انرژی مصرف شده توسط دستگاه ها (بر حسب وات ساعت).
۳. درصد دسترسی یا Availability: درصد زمانی که سیستم آنلاین و قابل استفاده می باشد. (بر حسب درصد)
۴. دقت یا Accuracy: درصد صحت داده های جمع آوری شده از سنسورها

گام های استفاده از مدل

نرمال سازی KPIها

چون شاخص ها واحدهای متفاوتی دارند، ابتدا باید همه مقادیر را به یک بازه یکسان به عنوان مثال ۰ و ۱ تبدیل کنیم. این کار باعث می شود تا همه KPIها بتوانند بدون واحد باشند. به عنوان مثال اگر زمان تاخیر بین ۰ تا ۱۰۰ میلی ثانیه است، مقدار ۵۰ میلی ثانیه به ۰/۵ نرمال سازی شود. یا اگر میزان دسترسی بین ۰ تا ۱۰۰ درصد است، مقدار ۸۰ درصد به ۰/۸ نرمال سازی شود.

وزن دهی

براساس اهمیت هر KPI یک وزن w_i به آن اختصاص می دهیم:

- $w_{1accuracy} = 0.4$
- $w_{2latency} = 0.3$
- $w_{3energy} = 0.2$
- $w_{4availability} = 0.1$

این وزن ها نشان دهنده اولویت و اهمیت KPI مورد نظر ما برای سیستم IoT می باشد.

ارزیابی عملکرد سیستم مذکور

با استفاده از این مدل همانطور که قبلاً گفته شد می‌توان سیستم‌های IoT را با وجود KPI های مختلف با یکدیگر مقایسه کرد: در فرمول ۳.۳ پارامتر $f(p_i)$ مقدار نرمال‌سازی شده هر KPI می‌باشد. مقادیر KPI های مورد نظر برای این سیستم پایش سلامت به صورت زیر می‌باشد:

• $f(p_1)$: 0.9 دقت

• $f(p_2)$: 0.6 زمان تاخیر

• $f(p_3)$: 0.7 مصرف انرژی

• $f(p_4)$: 0.8 میزان دسترسی

لازم به ذکر است ه مقادیر $f(p_i)$ توسط مدل‌هایی که تاکنون توضیح داده شد بدست آمده است و سپس مقادیر آن‌ها بین ۰ و ۱ نرمال‌سازی شده است. با توجه به مدل ۳.۳ خواهیم داشت:

$$(0.8 \times 0.1) + (0.7 \times 0.2) + (0.6 \times 0.3) + (0.9 \times 0.4) = 0.78 \quad (36)$$

برای سیستم مورد نظر با توجه به مقادیری که برای KPI های مورد نظر بدست آمده بود مقدار 0.78 در حقیقت مقدار مدل سیستم پایش سلامت شد. با استفاده از این مقدار می‌توان عملکرد سیستم‌های پایش سلامت دیگر (یا دیگر سیستم‌های IoT مربوط به آن دامنه) را نیز به همین ترتیب ارزیابی کرد و با مقادیری که در نهایت با استفاده از مدل ۳.۳ بدست می‌آید، هر کدام از آن‌ها را با یکدیگر مورد مقایسه و Trade off قرار داد.

دو KPI در سیستم‌های IoT وجود دارد که می‌تواند سطح کارایی آن‌ها را مشخص کند. اول تاخیر در سرویس‌دهی، دوم مصرف انرژی. به همین ترتیب برای محاسبه سطح کلی کارایی سیستم‌های IoT از مجموع مقادیر تابع سودمندی برای هر KPI استفاده می‌شود:

$$IoT_{sys}(p) = w_{T_{exe}} \times f(T_{exe}) + w_{e_{dev}} \times f(E_{dev}) \quad (37)$$

توابع $f(T_{exe})$ و $E_{con.}$ توابع سودمندی هستند برای مدت زمان اجرا و مصرف انرژی و در ادامه آن $w_{T_{exe}}$ و $w_{E_{conn.}}$ وزن‌های ضریب توابع هستند.

۴.۳ مدل‌سازی توابع سودمندی برای هر KPI

در این بخش بررسی می‌کنیم که چگونه می‌توان توابع سودمندی یا Utility functions را برای هر KPI مدل‌سازی کنیم. برای KPI های مانند تاخیر شبکه، مصرف انرژی، سرعت انتقال داده از توابع سیگموئیدی استفاده می‌شود که محدوده خروجی آن‌ها بین ۰ و ۱ می‌باشد و معیار عملکرد KPI را به صورت بی‌واحد نمایش می‌دهد. نکات مهم:

• $f(p_t)$ یا $f(p_i \uparrow)$: برای KPI هایی که مقدارهای بیشتر بهتر است مانند طول عمر باتری یا گذردهی.

• $f(p_u)$ یا $f(p_i \downarrow)$: برای KPI هایی که مقدار کمتر بهتر است مانند تاخیر یا مصرف انرژی.

۵.۳ فرمول تابع سیگموئید

$$f(p_i \uparrow) = L \left(\frac{A}{1 + e^{-\alpha_k \frac{(p_i - r_i)}{z_i}}} - U \right) \quad (38)$$

$$f(p_i \downarrow) = 1 - L \left(\frac{A}{1 + e^{-\alpha_k \frac{(p_i - r_i)}{z_i}}} - U \right) \quad (39)$$

$$L = \frac{1 + e^{\frac{\alpha_k \times r_i}{z_i}}}{e^{\frac{\alpha_k \times r_i}{z_i}}}. U = \frac{1}{1 + e^{\frac{\alpha_k \times r_i}{z_i}}} \quad (40)$$

- α_k : شیب منحنی است که حساسیت را به تغییرات KPI نشان می‌دهد (شب بیشتر برابر است با تغییر سریع‌تر عملکرد).
- r_i : نقطه مرکز یا مرجع KPI است (مثلاً یک مقدار استاندارد بهینه برای KPI که IoT باید به آن نزدیک باشد).
- L : بیشترین مقدار سودمندی
- U : کمترین مقدار سودمندی

۶.۳ اهداف فرمول‌های سیگموئید

- برای KPI‌هایی که افزایششان مطلوب است مانند گذردهی بیشتر بایستی $f(p \uparrow)$ را ماکزیمم کنیم.
- برای KPI‌هایی که کاهششان مطلوب است باید $f(p \downarrow)$ را مینیمم کنیم.

این مدل به ما اجازه می‌دهد که عملکرد KPI‌های مختلف را بدون واحد قابل مقایسه کنیم و KPI‌ها را براساس وزنی که به آن‌ها داده‌ایم مرتب کنیم و تصمیم‌گیری‌های بهینه‌تری برای بهبود عملکرد IoT انجام دهیم (برای مثال کدام KPI‌ها بیشتر نیاز به بهینه‌سازی دارند). روشی که در بالا ذکر شد اندازه‌گیری انعطاف‌پذیری را برای پارامترها ارائه می‌دهد حتی زمانی که هیچ مقدار مرجعی برای دستگاه IoT نداشته باشیم. این مدل به دلیل بی‌واحد بودن مقادیر به KPI خاصی وابسته نمی‌باشد. با این روش امکان تنظیم تابع سودمندی براساس حساسیت پارامترهای فردی فراهم می‌شود.