

پیش‌بینی و تحلیل توان مصرفی سیستم عامل‌ها

شهرام عندلیبی^۱ و مهران خسروی^۲

^۱ گروه کامپیوتر، دانشگاه آزاد اسلامی بروجرد، shahramandalibi@gmail.com

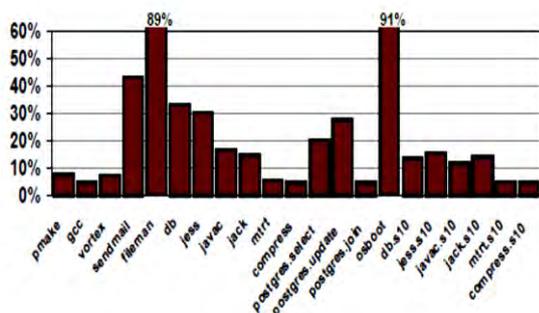
^۲ گروه کامپیوتر، دانشگاه آزاد اسلامی بروجرد، mehram_khosravy@yahoo.com

چکیده - در سیستم‌های کامپیوتری سیستم عامل بخش اصلی نرم‌افزاری سیستم است. با توجه به اینکه توان به عنوان عنصر اصلی هزینه در طراحی سیستم‌ها تبدیل شده است پیش‌بینی میزان توان مصرفی سیستم‌عامل‌ها از اهمیت ویژه‌ای برخوردار می‌باشد. چرا که با پیش‌بینی و مدل‌سازی توان سیستم‌عامل می‌توان پیش‌بینی دقیقی از توان مصرفی کل سیستم را نیز در اختیار داشت. برای اینکه بتوانیم توان را تخمین بزنیم پارامترهای تأثیر گذار بر روی توان مصرفی زیر روتین‌های سیستم عامل را مشخص کرده ایم. با استفاده از این عامل‌ها مجموعه داده‌هایی را برای عملیات داده کاوی تولید کرده‌ایم. برای پیش‌بینی توان و انجام عملیات داده کاوی از شبکه‌ی عصبی پرسپترون استفاده کرده‌ایم. طراحی این شبکه از الگوریتم بازگشت به عقب استفاده شده است که مدل‌سازی را از داده‌های آموزشی مستقل می‌کند. به این معنا که جامعیت مدل تولید شده بالا است. نتایج ارزیابی بیان‌گر کارایی بالای مدل پیشنهادی و میانگین خطای پیش‌بینی کمتر از ۱٪ است.

کلیدواژه- مدل‌سازی، توان مصرفی، شبکه عصبی، Softwatt

۱- مقدمه

افزایش محدودیت در مورد توان مصرفی به یک چالش در بسیاری از سیستم‌های کامپیوتری تبدیل شده است، که نیاز به پیش‌بینی و تخمین توان مصرفی را ضروری ساخته است. نرم افزارها، که به صورت نرم افزارهای سیستمی و نرم افزارهای کاربردی قابل دسته‌بندی هستند، بخش اصلی سیستم‌های امروزی را تشکیل می‌دهند. اجرای نرم‌افزار موجب فعال شدن سخت افزار می‌شود و شیوه‌ای که نرم‌افزار از سخت‌افزار استفاده می‌کند تأثیر مستقیمی بر میزان توان مصرفی سیستم دارد [۱]. سیستم‌عامل نه تنها بخش مهمی از عملکرد کلی سیستم را در برمی‌گیرد بلکه می‌تواند بخش اصلی توان مصرفی سیستم را نیز به خود اختصاص دهد. برای نمونه، شکل ۱ درصد کل توان مصرفی (ریزپردازنده و حافظه) توسط سیستم‌عامل در یک برنامه آزمایشی را نشان می‌دهد. نادیده گرفتن اثرات سیستم‌عامل می‌تواند منجر به خطای تخمین توان در پیش‌بینی شود [۲].



شکل ۱- درصد توان مصرفی سیستم عامل در یک برنامه آزمایشی

بنابراین، در محاسبه توان مصرفی سیستم‌عامل بهتر است که پیش‌بینی در سایه بارکاری شدید سیستم صورت پذیرد. در سیستم‌های دارای محدودیت توان چگونگی بکارگیری فضای حافظه برای طراحان سیستم اهمیت دارد چرا که عامل موثری در میزان توان مصرفی برنامه‌ها است [۲]. در ادامه این مقاله از بخش‌های زیر تشکیل شده است: در بخش ۲ به انواع مدل‌های تخمین توان پرداخته شده است. در بخش ۳ به بررسی کارهای مرتبط پرداخته‌ایم. موضوع بخش ۴ ساختار مدل پیشنهادی است. بخش ۵ به ارزیابی میزان کارایی مدل پیشنهادی پرداخته است. بخش ۶ این مقاله نیز به

توضیح می‌دهد که یک بخش کلیدی در شبیه سازی سطح چرخه است.

۲-۱-۲ مدل توان سطح دستورالعمل

در مقابل روش‌های سطح چرخه، روش‌های تحلیل توان سطح دستورالعمل وجود دارد. در این روش وقتی دستورالعمل‌های برنامه اجرا می‌شوند کل توان مصرف شده، محاسبه می‌کند [۶]. هزینه‌های توان دستورالعمل به دستورالعمل هزینه‌ی مبنا نامیده شده که می‌تواند برای دستورالعمل‌های یک پردازنده خاص اندازه‌گیری شوند. مصرف توان قابل توجهی ناشی از سویچ دستورالعمل‌های میانی متوالی وجود دارد که بیشتر توسط پایپ‌لاین و حافظه‌ی نهان صرف می‌شوند. توان پایه‌ی دستورالعمل‌ها و توان سویچ بین دستورالعمل‌ها براساس روند تجربی با استفاده از یک برنامه که چندین نمونه از دستورالعمل هدف (برای ارزیابی هزینه‌ی پایه) را دارا می‌باشد اندازه‌گیری می‌شود.

وقتی که مدل توان دستورالعمل به دستورالعمل برای پردازنده‌ای خاص ساخته شد، کل انرژی، E_p توسط فرمول زیر محاسبه می‌شود:

$$E_p = \sum_i (B_i * N_i) + \sum_{ij} (O_{ij} * N_{ij}) + \sum_k S_k \quad (1)$$

که در آن B_i بهترین توان برای پردازش دستورالعمل i می‌باشد. O_{ij} نماینده توان ائتلافی در اثر سویچینگ مدار بین هر جفت دستورالعمل اجرا شده (i, j) متوالی می‌باشد. پارامتر S_k نشان دهنده باقی توان مصرفی در اثر k - نمونه از اثرات درون دستورالعملی از قبیل نوشتن بافر و حافظه نهان می‌باشد. برای یک برنامه مشخص، کل مصرف انرژی، E ، را می‌توان توسط ضرب کردن B_i و O_{ij} با نمونه‌های پویای هر یک از دستورات (N_i) و جفت دستورالعمل‌ها (N_{ij}) به ترتیب محاسبه نمود [۷].

۲-۲-۲ مدل های توان سطح سیستم

بررسی توان پردازنده برای تخمین توان کلی سیستم کافی نیست. اگر اجزای دیگر که تاثیر قابل توجهی بر مصرف توان داشته باشند را در نظر بگیریم مدل جامع‌تر و بهتری را می‌توانیم ارائه بدهیم. بنابراین، لازم است که اجزای مهم دیگر را برای مدل کردن توان کل سیستم در نظر بگیریم. این بخش مدل وضعیت سطح سخت افزار را مطرح می‌کند که مصرف کل توان سیستم براساس وضعیت یا حالت هر بخش سخت‌افزاری ارزیابی می‌کند. در اینجا، فرض شده است که هر قسمت قادر

نتیجه‌گیری کلی از مدل پیشنهادی و مباحث مطرح شده در این مقاله پرداخته است.

۲- روش های مدل سازی و تخمین توان

توان به عنوان یک محدودیت مهم در طراحی پردازنده‌ها و سیستم‌های کامپیوتری تبدیل شده است. اولین مرحله برای مدل سازی مصرف توان، شناسایی منابع مصرف کننده توان در سطوح مختلف سیستم است [۲]. مدل‌های تخمین توان گوناگونی طراحی شده‌اند. بخش ۱-۲ مدل‌های توان مصرفی پردازنده را توضیح داده که مصرف توان را در چرخه یا سطح دستورالعمل ارزیابی می‌کند. بخش ۲-۲ مدل‌های توان سیستم را مطرح کرده که مصرف توان را در اجزای نرم‌افزاری و سخت‌افزاری بررسی می‌کند.

۲-۱-۲ مدل های توان مصرفی سطح پردازنده

توانی که توسط پردازنده مصرف می‌شود بخش مهمی از کل توان مصرفی سیستم‌های کامپیوتری است. بنابراین هدف اصلی تحلیل توان این قسمت می‌باشد [۳]. چندین مدل برای مدل سازی توان مصرفی پردازنده‌ها طراحی شده است که در شبیه‌سازی‌های موجود به منظور ارزیابی مصرف توان پردازنده‌ها مورد استفاده قرار می‌گیرند. این تحلیل‌ها براساس چرخه (یا سطح انتقال ثبات) و دستورالعمل هستند که به ترتیب در دو بخش زیر توضیح داده می‌شوند.

۲-۱-۱-۲ مدل توان سطح چرخه

مصرف توان پردازنده را می‌توان با استفاده از ساختار شبیه‌سازی سطح چرخه ارزیابی کرد. شبیه‌سازی‌های توان مصرفی سطح معماری نشان داده‌اند که برای پردازنده‌هایی با پایپ‌لاین عمیق، اجرای خارج از ترتیب (out-of-order) و هوشمند قابل استفاده می‌باشند [۴].

این کار توسط تعیین بخش‌های فعال (یا مشغول) هر چرخه‌ی اجرایی پردازنده شبیه‌سازی شده، انجام می‌شود که بخش‌های فعال در حقیقت ساختارهای مجزای موجود در پردازنده می‌باشند [۵]. این روش یعنی، استفاده از منابع چرخه به چرخه می‌تواند برای ارزیابی مصرف توان استفاده شود. این روش مدل‌سازی توان چگونگی استفاده از توان هر بخش را

۲-۲-۳ مدل توان ترکیبی

روش‌هایی که در بالا ذکر شد توان مصرفی را برای اجرای یک برنامه‌ی خاص ارائه می‌دهند، اما بدون درک حالت‌های کل سیستم با جزئیات کافی بین همه‌ی اجزای سیستم اقدام به این کار می‌کنند. شبیه‌ساز SoftWatt [۱۱] با مدل سازی اجزای سخت‌افزاری مانند پردازنده، سلسله مراتب حافظه و سیستم دیسک و تعیین توان برنامه بر این مشکل غلبه می‌کند. SoftWatt روی زیرساخت SimOS [۱۲] ساخته شده است. برای به دست آوردن توان کامل سیستم، SoftWatt مدل‌های توان تحلیلی مختلفی را در دسترس شما قرار می‌دهد. بخش‌های نمونه در SoftWatt شامل مسیر داده، حافظه و گرداننده‌ی دیسک سخت می‌باشد. از لحاظ سخت‌افزاری، دیسک بزرگترین مصرف کننده‌ی توان کل سیستم است. با این وجود، با اتخاذ دیسک با توان پایین، نقطه بحرانی مصرف توان به پردازنده تغییر پیدا می‌کند. از دیدگاه نرم‌افزاری دستورات کاربر توان بیشتری نسبت به دستورات هسته اصلی مصرف می‌کند. با این وجود سرویس‌های اصلی آنهایی هستند که توان قابل توجهی در پردازنده و سلسله مراتب حافظه مصرف می‌کنند. بنابراین، مصرف توان دستورات هسته‌ی اصلی را در نظر می‌گیرد که برای کاهش هزینه کل توان مهم است. پردازنده و زیرسیستم حافظه برای کاهش توان مصرفی دستوراتی نظیر متوقف کردن پردازنده را اجرا می‌کنند. این دستورات در زمانی که پردازنده در انتظار اجرای دستورات است، به طور قابل توجهی می‌تواند مصرف توان را کاهش دهد.

۲-۲-۳-۳ چالش‌های موجود در مدل‌های تخمین توان

- **سرعت بالا:** تخمین توان باید از کند کردن چرخه اجرای دستورات اجتناب کند.
- **توانایی تخمین زمان اجرا و سربار کمی داشته باشد:** مدل باید بتواند تخمین را در حال اجرای برنامه انجام دهد و تخمین توان مصرفی زمان اجرا باید سربار خیلی کم داشته باشد که از وقفه در عملکرد نرمال سیستم اجتناب نماید.
- **سادگی، در دسترس بودن، قابلیت تعمیم:** مدل باید بر تعداد کمی از پارامترهای توان متکی بوده که بطور وسیعی در دسترس باشند که در ساختارهای سخت‌افزاری مختلف قابل دسترسی باشند.

به تغییر به یکی از چندین وضعیت ممکن می‌باشد برای مثال وضعیت حالت معوق (حالتی است که عملیات بدون آنکه به آن پایان داده شود معوق می ماند و در حافظه باقی می ماند تا در اثر روی دادن رخدادی مجدد فعال شود یا به اصطلاح بیدار شود) می‌تواند به وضعیت فعال تغییر پیدا کند [۸]. این قابلیت معمولاً در سیستم‌های قابل حمل بکار برده می شود چرا که بازه زمانی عملکرد فعال این مدل سیستم‌ها بسیار با اهمیت است.

۲-۲-۲ مدل توان سخت‌افزاری

Cignetti و همکاران مدل توان سیستم را با مدل توان وضعیت پایه‌ی سخت‌افزار ارائه داده‌اند [۸]. این مدل توان جزئیات سطح پایین هر زیرسیستم سخت‌افزار را با تعیین مجموعه‌ی حالت‌های توان برای هر قسمت (برای مثال، حالت معوق، یا پرتراکم برای پردازنده) در نظر می‌گیرد. هر حالت توان با مصرف توان سخت‌افزار طی حالتی که توان حالت پایدار نامیده می‌شود، مشخص می‌شود. علاوه بر این، هر انتقال بین حالت‌ها، هزینه‌ی مصرف توان را تعیین می‌کند که جابه‌جایی توان نامیده می‌شود. می‌توان توانی مشابه را برای حفظ مسیر اجرایی دستورات سیستم بکار ببریم زیرا انتقال بین حالت‌ها در نتیجه دستورات سیستم رخ می‌دهد. کل توان مصرف شده توسط سیستم با اضافه کردن توان حالت هر قسمت، ضربدر زمان صرف شده در آن حالت به اضافه‌ی مصرف کل توان برای همه انتقال‌ها تعیین می‌شود.

۲-۲-۲-۲ مدل توان براساس پردازش‌ها

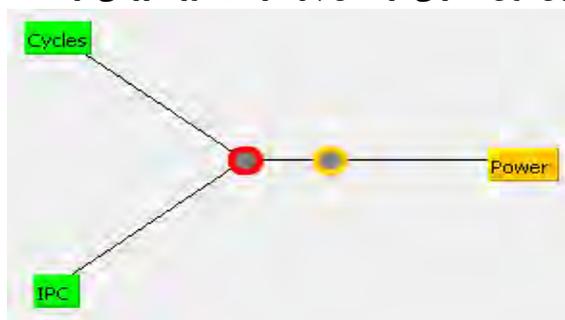
از آنجایی که نرم‌افزار تعیین کننده‌ی فعالیت‌های اصلی اجزای سخت‌افزار مانند هسته پردازنده، حافظه و گذرگاه سیستم است، برای ارزیابی توان به روش‌های نرم‌افزاری و فعل و انفعال آنها نیاز است. این بخش روش اندازه‌گیری توان براساس پردازش‌ها برای سیستم است [۹ و ۱۰]. این روش اساساً بر اندازه‌گیری مصرف توان کل سیستم مورد نظر قرار داده و سعی دارد تمرکز را بر برنامه‌های کاربردی و روند عملکردی سیستم نشان دهد. این روش براساس پردازش است به این معنی که پردازش‌های مختلفی را در نظر می‌گیرد که نه تنها توان مصرفی متفاوت، بلکه مدت زمان و تعداد دستورات عمل‌های متفاوتی را اجرا می‌کنند. البته باید این نکته را نیز در نظر داشت که از مجموعه‌های مختلفی از منابع در ترتیب‌های مجزا استفاده می‌کنند.

۳- کارهای مرتبط

یک شبکه چند لایه از روش Back Propagation استفاده می شود. در این روش با استفاده از Gradient Descent سعی می شود تا مربع خطای بین خروجی های شبکه و تابع هدف کمینه شود.

در طراحی شبکه عصبی از ۵۵ زیر روتین های سیستم عامل استفاده شده است. مقادیر توان این زیر روتین ها توسط شبیه ساز SoftWatt اندازه گیری شده است. این مقادیر حاصل شده از اجرای برنامه های استاندارد در محیط شبیه ساز SoftWatt است. با توجه به اینکه هر زیر روتین توانی مخصوص به خود را مصرف می کند می توان با مدل کردن توان مصرفی این زیر روتین ها توان کلی مصرفی برنامه ها را مدل نمود چراکه برنامه ها برای اجرا از این زیر روتین ها استفاده می کنند.

هر زیر روتین توسط دو پارامتر مورد بررسی قرار گرفته است: ۱- IPC - ۲ Cycle. سپس این زیر روتین ها به منظور آموزش و استخراج الگو مورد داه کاوی قرار می گیرند. برای استخراج ارتباط بین پارامترها از شبکه های عصبی استفاده شده است. این شبکه ها به دلیل قابلیت یادگیری بسیار بالایی که دارند به خوبی می توانند با آموزش، ارتباط بین پارامترها و میزان توان مصرفی را تشخیص داده و مدل جامعی را برای تخمین میزان توان مصرفی براساس پارامترهای مورد بررسی ارائه دهند.



شکل ۲- نمونه شبکه عصبی طراحی شده برای تخمین توان بخشی از مجموعه داده استفاده شده در جدول ۴ نشان داده شده است.

Dick و همکاران روشی را برای محاسبه توان سیستم عامل در سیستم های تعبیه شده بر پایه ی مدل سطح چرخه پردازنده ارائه دادند. مدل پیشنهادی آنها دارای خطای ۱۳ درصد بود. از جمله معایب آن نیز می توان به عدم جامعیت مدل اشاره نمود [۱۳].

Lu و همکاران مدلی را بر مبنای رگرسیون ارائه دادند که توان سیستم عامل را در محیط های تعاملی با خطایی بیشتر از ۱۰ درصد محاسبه می کرد. مشکل مدل پیشنهادی زمان بر بودن مراحل مدل سازی بود [۱۴].

Healy و همکاران مدلی برای سیستم های سنسور ارائه دادند که توان مصرفی سیستم عامل در این محیط ها را با خطایی کمتر از ۱۸ درصد تخمین می زد. از جمله مشکلات مدل پیشنهادی حساس بودن به پلت فرم اجرایی و عدم جامعیت را می توان ذکر کرد [۱۵].

۴- مدل پیشنهادی

هدف تخمین توان در سیستم عامل ها براساس پارامترهای اصلی تعیین کننده در توان است. به این منظور از شبکه عصبی و نرم افزار داده کاو Weka [۱۶] استفاده شده است. شبکه عصبی روشی برای محاسبه است که بر پایه ی به هم پیوسته چندین واحد پردازشی ساخته می شود. شبکه از تعداد دلخواهی سلول یا گره یا واحد یا نرون تشکیل می شود که مجموعه ورودی را به خروجی ربط می دهند و برای مدل سازی مسائل مختلفی بکارگیر می رود.

مرسوم ترین نوع شبکه عصبی پرسپترون می باشد. این نوع از شبکه عصبی از طبقه شبکه های ناظر می باشد دلیل این امر نیز این مسئله می باشد که ابتدا باید هدف و خروجی سیستم مشخص شود. هدف از این نوع شبکه ایجاد یک مدلی است که با گرفتن ورودی، خروجی را بطور صحیحی تولید نماید که این کار را با استفاده از اطلاعات پیش فرض شبکه انجام می دهد. این شبکه می تواند به صورت تک لایه و یا چند لایه وجود دارد. پرسپترون تک لایه تنها می تواند مسایل مجزای خطی را دسته بندی کند و برای مسایل پیچیده تر لازم است که از تعداد بیشتری لایه استفاده کنیم. پرسپترون چند لایه یک شبکه کاملاً بهم مرتبط می باشد چرا که هر نرون در یک لایه به تمامی نرون های لایه بعدی مرتبط می باشد. برای یادگیری وزن های

سازی است چیزی که تا به امروز کمتر در مدل سازی توان مصرفی سیستم عامل ها مورد استفاده و توجه قرار گرفته است .

```

=== Evaluation on training set ===
=== Summary ===
Correlation coefficient          0.959
Mean absolute error            0.9727
Root mean squared error        1.216
Total Number of Instances      55
  
```

شکل ۳- نتایج ارزیابی مدل پیشنهادی

```

=== Predictions on training set ===
inst#  actual  predicted  error
1      45     45.073    0.073
2      33     31.813   -1.187
3      35     32.574   -2.426
4      27     25.417   -1.583
5      36     36.72     0.72
6      33     32.232   -0.768
7      39     39.418    0.418
8      25     25.059    0.059
9      18     18.296    0.296
10     45     45.766    0.766
  
```

شکل ۴- نمونه ای خروجی مدل پیشنهادی

۶- نتیجه گیری

سیستم های کامپیوتری به علت عملکرد بالای پردازنده های چند منظوره و نرم افزارهای (سیستم عامل و برنامه های کاربردی) در حال اجرا بر روی آنها توان مصرفی قابل توجهی را مورد استفاده قرار می دهند. تخمین توان بطور قابل توجهی به یک مسئله اصلی در طراحی سیستم ها تبدیل شده است.

روشی که برای مدل سازی توان مصرفی سیستم عامل در این مقاله پیشنهاد شده است اساسا بروی مفهوم داده کاوی و یادگیری الگو قرار دارد و قابلیت استفاده در سیستم هایی را دارد که پارامترهای مدنظر را داشته باشند این امر باعث افزایش جامعیت مدل پیشنهادی نسبت به سایر مدل های مشابه در این زمینه است با تحلیل پارامترهای انتخاب شده این امکان فراهم می شود که راهکارهایی برای کاهش توان مصرفی برنامه با ایجاد تغییر بر روی پارامترها پیدا کرد.

نتایج ارزیابی بیان گر کارایی بالای مدل پیشنهادی است چراکه میانگین خطا کمتر از ۱ است. این میزان خطا با توجه به جامعیت مدل و تعداد پارامترهایی که برای تخمین مورد استفاده قرار گرفته اند بسیار مطلوب است برای اینکه بتوان جامعیت مدل را افزایش داد راهکارهایی نظیر افزایش تعداد پارامترها در فرایند مدل سازی و یا اختصاصی کردن داده های آموزشی به یک سیستم خاص را می توان در نظر داشت.

جدول ۱- نمونه داده های استفاده شده برای ساخت مدل پیشنهادی

OS Services	Avg Cycles	IPC Avg	Power Avg
read	2,614	1.36	45
write	9,344	0.91	33
open	8,626	0.97	35
close	2,131	0.77	27
unlink	8,904	1	36
lseek	536	1.01	33
access	6,547	1.11	39
dup	1,074	0.74	25
ioctl	5,230	0.51	18
fcntl	613	1.39	45
getdents	5391	1	34
xstat	5,990	1.22	43
lxstat	3,517	1.52	53
fxstat	1,293	0.85	28
utlb	13	0.92	28
pfault	1,100	1.16	40
vfault	971	1.43	47
COW_fault	2,574	1.65	54
demand_zero	1,939	1.54	44
.....			
sigsuspend	9,901	0.3	11
getcontext	679	1.38	43
setcontext	1,025	0.97	32

هنگامی توان زیرروتین ها با استفاده از شبکه عصبی محاسبه شد، با توجه به تعداد اجرای زیرروتین ها می توان توان سیستم عامل را محاسبه نمود.

۵- ارزیابی

در زمینه تخمین توان مصرفی مدلی های مختلفی ارائه شده اند. برای آنکه بتوان کارهای مختلف را از نظر میزان کارایی با یکدیگر مقایسه نمود باید معیار یکسانی را تعریف نمود. در بیشتر مقالات از MRE (Mean Absolute Error) و RMSE (Root Mean Squared Error) نیز برای نشان دادن تفاوت عملکرد دو تکنیک متفاوت استفاده کرد. روابط (۲) و (۳) روش به دست آوردن این دو مقدار است.

$$MRE = \frac{|a_1 - c_1| + |a_2 - c_2| + \dots + |a_n - c_n|}{n} \quad (2)$$

$$RMSE = \sqrt{\frac{(a_1 - c_1)^2 + (a_2 - c_2)^2 + \dots + (a_n - c_n)^2}{n}} \quad (3)$$

که در این دو رابطه a مقدار خروجی واقعی و c مقدار خروجی قابل قبول است. با این دو معیار می توان میزان خطای پیش بینی از مقدار واقعی را تعیین نمود و هر اندازه این مقدار کمتر باشد مدل توصیف شده عملکرد بهتری خواهد داشت.

در مدل سازی به این روش امکان تخمین نزدیک تر به واقعیت در زمان مواجه با رخداد های جدید فراهم می شود دلیل این ویژگی بهره گیری از مفهوم آموزش و یادگیری الگو در مدل



- [6] O. A. Patino and M. Jimenez, "Instruction Level Power Profile for the PowerPC Microprocessor," *Computing Research Conference 2003, University of Puerto Rico Mayagüez*, pp.120-123, April 2003.
- [7] V. Tiwari, S. Malik, A. Wolfe, and M. T. Lee "Instruction Level Power Analysis and Optimization of Software," *Journal of VLSI Signal Processing*, Vol. 13, Issue 2-3, August 1996.
- [8] T. Cignetti, K. Komarov and C. Ellis, "Energy Estimation Tools for the PalmTM," *Proceedings of the ACM MSWiM'2000: Modeling, Analysis and Simulation of Wireless and Mobile Systems*, August 2000.
- [9] F. Chang, K. Farkas and P. Ranganathan, "Energy-driven Statistical Profiling: Detecting Software Hotspots," *Proceedings of the Workshop on Power Aware Computing Systems*, February 2002.
- [10] J. Flinn and M. Satyanarayanan. "PowerScope: A tool for Profiling the Energy Usage of Mobile Applications," *Proceedings of the Workshop on Mobile Computing Systems and Applications (WMCSA)*, February 1999.
- [11] S. Gurumurthi, A. Sivasubramaniam, M. J. Irwin, N. Vijaykrishnan, and M. Kandemir, "Using Complete Machine Simulation for Software Power Estimation: The SoftWatt Approach," *Proceedings of the International Symposium on High Performance Computer Architecture (HPCA-8)*, February 2002.
- [12] SimOS, <http://simos.stanford.edu/>
- [13] Robert, P. Dick, Anand Raghunathan, and Niraj K. Jha, "Analysis of Power Dissipation in Embedded Systems Using Real-time Operating Systems", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 22, No. 5, May 2003. pp.615-627.
- [14] Y. H. Lu, L. Benini, G. D. Micheli, "Power aware operating systems for interactive systems", *IEEE Transactions on Very Large Scale Integration Systems*, Volume 10, Issue 2, April 2002, pp. 119-134
- [15] Healy, M.; Newe, T.; Lewis, E. Power management in operating systems for wireless sensor nodes. In *Proceedings of IEEE Sensors Applications Symposium*, San Diego, CA, USA, February 6-8, 2007; pp. 1-6.
- [16] H. Witten and E. Frank. *Data mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2005.

سپاسگزاری

از آقای دکتر صالحی و آقای دکتر بزاز از دانشگاه صنعتی شریف که ما را در ارائه این مقاله یاری کردند، کمال تشکر را داریم.

مراجع

- [1] D. Ofelt and J. L. Hennessy, Efficient Performance Prediction for Modern Microprocessors, In *Proceedings of the International Conference on Measurement and Modeling of Computer Systems*, 2000.
- [2] M. Huang, J. Renau, S. M. Yoo and J. Torrellas, A Framework for Dynamic Energy Efficiency and Temperature Management, In *Proceedings of the International Symposium on Microarchitecture*, 2000.
- [3] D. Brooks, P. Bose, S. Schuster, H. Jacobson, P. Kudva, A. Buyuktosunoglu, J. Wellman, V. Zyuban, M. Gupta and P. Cook, "Power Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors," *IEEE Micro*, pp. 26-44, December 2000.
- [4] Intel Pentium 4 Processors - Manuals, Intel Corporation, 2002.
- [5] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proceedings of the 27th International Symposium on Computer Architecture (ISCA)*, June 2000.