Journal of Iranian Association of Electrical and Electronics Engineers
Vol. 22 No. 1, 2025
DOI: 10.61186/jiaeee.22.1.125

Optimization of Power Management in Multi-Core Processors Using Machine Learning and Dynamic Voltage and Frequency Scaling

Farideh Ziaee[1] , Mahdieh Ghazvini[2] , Marjan Kaedi[3]

[1] MSc. Computer Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran, (ORCID: 0000-0001-6630-5315)
faridehziaee@eng.uk.ac.ir
[2] Associate Professor, Computer Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran, (ORCID: 0000-0002-3647-2101)
mghazvini@uk.ac.ir
[3] Associate Professor, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, (ORCID: 0000-0003-2000-4781)
kaedi@eng.ui.ac.ir

Abstract :

As transistor sizes in processors continue to shrink, overall energy consumption has paradoxically increased due to the growing number of transistors. This trend has led to significant thermal challenges and a decrease in system performance. Additionally, circuit aging has emerged as a major concern, negatively affecting both processor performance and longevity. Dynamic Voltage and Frequency Scaling (DVFS) is a widely adopted power management technique that mitigates energy consumption and improves system durability by dynamically adjusting the processor's voltage and frequency. This paper presents a novel machine learning-based approach for power management in multi-core processors. The proposed method leverages input feature analysis and combines a decision tree algorithm with DVFS techniques to accurately predict and allocate the optimal voltage and frequency for each core. Evaluation results demonstrate that the model achieves a prediction accuracy of 95%, effectively forecasting system performance across various workloads. This approach makes a significant contribution to the development of energy-efficient systems.

Keywords: **Multi-core processors, Power manager, Machine learning, Dynamic Voltage and Frequency Scaling (DVFS), Decision tree algorithm.**

Article Type: Research

Received: 11. 03. 2023

Revised: 25. 09. 2023

Accepted: 19. 11. 2024

Corresponding author: Mahdieh Ghazvini

Corresponding author's address: Shahid Bahonar University of Kerman- Saba Boulevard- Research Square- Kerman- Iran

Copyright © 2025 The Authors. Published by Iranian Association of Electrical and Electronics Engineers.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International license (https://creativecommons.org/licenses/by-nc/4.0/). Non-commercial uses of the work are permitted, provided the original work is properly cited.

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

1. Motivation of the work

Increasing the number of cores improves throughput and multi-threaded processing. However, the rise in power density causes only a limited number of transistors to be active simultaneously. This leads to the formation of hot spots and a reduction in system reliability [1-5]. Dynamic Voltage and Frequency Scaling (DVFS), as a key technique, reduces power consumption by dynamically adjusting voltage and frequency.

2. Contributions

This study presents a novel decision tree-based approach for accurately predicting key system performance parameters. This method utilizes DVFS technique and a linear regression algorithm to determine the optimal voltage and frequency for each core. Traditional approaches rely on manual configurations or fixed rules, while our proposed method dynamically adapts to workload changes and makes appropriate decisions. The low computational complexity and high learning speed are additional advantages of this method. The decision tree algorithm is chosen for power management due to its ability to select key features and make precise decisions at each stage. Furthermore, linear regression facilitates voltage and frequency allocation by reducing computational complexity and improving speed. The aim of this study is to improve power management and increase efficiency in processing systems.

3. Procedures

To implement and evaluate the proposed method, the Gem5 simulator in full system mode is used along with the McPAT tool. This tool was employed to calculate the metrics such as power consumption, energy, execution time, and processing performance.

Initially, the data were classified using the decision tree algorithm to predict the system state. Then, linear regression was applied to allocate optimal voltage and frequency values to each processor core. The models were evaluated using metrics such as accuracy, precision, recall, and F1 score (the harmonic mean of precision and recall). Analysis of variance (ANOVA) was conducted to evaluate the algorithm's performance after predicting the system states and allocating voltage and frequency.

4. Findings

As mentioned, this study utilized the decision tree algorithm and linear regression to predict the optimal voltage and frequency levels for multi-core processors. The decision tree was chosen due to its simplicity, interpretability, and high efficiency in data classification. In addition to the decision tree, the results were also compared with two other algorithms: the Naive Bayes classifier [6] and the K-NN algorithm [7]. The evaluation results showed that the decision tree yielded better results. The evaluation of the algorithms was conducted using the metrics of accuracy, precision, recall, and F1 score, and it was found that the decision tree performed better, with results of 92, 93.7, 97.83, and 95.7 for these four metrics. After predicting the system states and allocating voltage-frequency pairs, analysis of variance (ANOVA) was used to evaluate the algorithm's performance. From this analysis, the following results can be inferred:

Sum of Squared Errors (SSE): To assess the model's accuracy, the difference between the actual and predicted values was calculated. The values of 0.033 and 0.005 for the dependent variables indicate that the error is very small and close to zero.

Coefficient of Determination (R^2): The values of 0.17 and 0.08 indicate that the model explains 20% and 10% of the variance in the dependent variables, respectively. The closer these values are to 1, the better the model can explain the changes in the dependent variable.

Adjusted R^2: The obtained adjusted R^2 values were 0.15 and 0.06 which suggests that the independent variables in the model are effective in explaining the changes in the dependent variable.

F-Ratio: The F-ratio, calculated by dividing MSR by MSE, shows values of 9.6 and 4.05. The magnitude of these values indicates that the regression model effectively explains the total variance compared to the residuals.

5. Conclusion

This study presents an approach for power management in multi-core processors using DVFS techniques and machine learning methods. The method, leveraging classification algorithms, has been able to accurately predict the optimal voltage and frequency levels for various workloads. The application of this approach resulted in reduced computational overhead and improved overall system efficiency. The results demonstrated that this method adapts well to system changes and different workloads, and by optimizing decisions based on historical data, it contributes to enhanced performance and increased system lifetime.

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

نشریه مهندسی برق و الکترونیک ایران

نشریه مهندسی برق و الکترونیک ایران، سال بیست و دوم، شماره اول، بهار ۱۴۰۴، صفحات ۱۲۵-۱۳۵

DOI: 10.61186/jiaeee.22.1.123

بهینه‌سازی مدیریت توان در پردازنده‌های چندهسته‌ای با یادگیری ماشین و مقیاس‌بندی پویا ولتاژ و فرکانس

فریده ضیایی[۱]، مهدیه قزوینی[۲]، مرجان کائدی[۳]

۱- دانشجوی کارشناسی ارشد- بخش مهندسی کامپیوتر- دانشکده فنی و مهندسی- دانشگاه شهید باهنر کرمان
faridehziaee@eng.uk.ac.ir

۲- دانشیار- بخش مهندسی کامپیوتر- دانشکده فنی و مهندسی- دانشگاه شهید باهنر کرمان
mghazvini@uk.ac.ir

۳- دانشیار- دانشکده مهندسی کامپیوتر- دانشگاه اصفهان
kaedi@eng.ui.ac.ir

چکیده:

با کوچک‌تر شدن ترانزیستورها در پردازنده‌ها، مصرف انرژی کلی آن‌ها به‌جای کاهش، به دلیل افزایش تعداد ترانزیستورها افزایش‌یافته است. این امر منجر به مشکلات حرارتی و کاهش کارایی کلی سیستم شده است. علاوه بر این، سالخوردگی مدارها نیز اهمیت زیادی پیدا کرده است؛ زیرا بر عملکرد و طول عمر پردازنده‌ها تأثیر منفی می‌گذارد. مقیاس‌بندی پویای ولتاژ و فرکانس، روشی برای مدیریت توان است که با تغییر ولتاژ و فرکانس پردازنده، مصرف انرژی را کاهش داده و طول عمر سیستم را بهبود می‌بخشد. در این مقاله، یک روش مبتنی بر یادگیری ماشین برای مدیریت توان در پردازنده‌های چندهسته‌ای معرفی می‌شود. این روش با بهره‌گیری از تحلیل ویژگی‌های ورودی و استفاده از الگوریتم درخت تصمیم‌گیری و تکنیک مقیاس‌بندی پویا ولتاژ و فرکانس، قادر است سطح بهینه‌ی ولتاژ و فرکانس هر هسته را پیش‌بینی و به‌صورت دقیق تخصیص دهد. نتایج ارزیابی نشان می‌دهد که مدل پیشنهادی با دقت ۹۵ درصد قادر به پیش‌بینی عملکرد سیستم در شرایط کاری مختلف است و می‌تواند به‌طور مؤثر در طراحی سیستم‌های کم‌مصرف به کار گرفته شود.

کلمات کلیدی: پردازنده‌های چندهسته‌ای، مدیر توان، یادگیری ماشین، مقیاس‌بندی پویای ولتاژ و فرکانس، الگوریتم درخت تصمیم‌گیری

نوع مقاله: پژوهشی

دریافت: ۱۴۰۱/۱۲/۲۰

بازنگری: ۱۴۰۲/۰۷/۰۳

پذیرش: ۱۴۰۳/۰۸/۲۹

نام نویسنده‌ی مسئول: دکتر مهدیه قزوینی

نشانی نویسنده‌ی مسئول: ایران - کرمان - میدان پژوهش - بلوار صبا - دانشگاه شهید باهنر کرمان - بخش مهندسی کامپیوتر

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

حق انتشار این مستند، متعلق به نویسندگان آن است. ۱۴۰۴ ©. ناشر این مقاله، انجمن مهندسین برق و الکترونیک ایران است.

این مقاله تحت گواهی زیر منتشرشده و هر نوع استفاده غیرتجاری از آن مشروط بر استناد صحیح به مقاله و با رعایت شرایط مندرج در آدرس زیر مجاز است.

Creative Commons Attribution-NonCommercial 4.0 International license
(https://creativecommons.org/licenses/by-nc/4.0/)

۱- مقدمه

ریزپردازنده‌های چندهسته‌ای با پیشرفت در کوچک‌سازی ترانزیستورها و نوآوری‌های معماری کامپیوتر توسعه‌یافته‌اند. افزایش تعداد هسته‌ها در این پردازنده‌ها به بهبود توان عملیاتی و پردازش چندرشته‌ای منجر می‌شود. بااین‌حال، این پیشرفت‌ها با چالش‌هایی نظیر افزایش توان مصرفی، مشکلات حرارتی و کاهش طول عمر مدارها همراه بوده است. افزایش چگالی توان و مشکلات حرارتی به این معنی است که تنها تعداد محدودی از ترانزیستورها می‌توانند به طور هم‌زمان فعال باشند و این تعداد در نسل‌های جدید کوچک‌تر می‌شود که ممکن است به نقاط داغ، پیری غیریکنواخت، تسریع در شکست تراشه و کاهش قابلیت اطمینان منجر شود [۳-۷].

برای مقابله با این چالش‌ها، روش‌های مختلفی برای افزایش طول عمر سیستم‌ها پیشنهاد شده است. این روش‌ها به دودسته کلی تقسیم می‌شوند: روش‌های سطح پایین و روش‌های سطح بالا. روش‌های سطح پایین شامل بهینه‌سازی در سطح ترانزیستورها و مدارهای مجتمع هستند که به کاهش مصرف توان و بهبود عملکرد مدارهای VLSI[1] کمک می‌کنند [۱]. در مقابل، در روش‌های سطح بالا، مدیریت توان در سطح سیستم به‌عنوان نقطه شروع طراحی مورداستفاده قرار می‌گیرد تا با تنظیم پویای ولتاژ و فرکانس، صرفه‌جویی بهینه در توان و بهبود کارایی سیستم حاصل شود [۲]. یکی از تکنیک‌های کلیدی در این زمینه، مقیاس بندی پویای ولتاژ و فرکانس(DVFS) [2] است که به کاهش توان مصرفی و کنترل گرمای تولید شده کمک می‌کند [۲,۲۷].

در کنار DVFS، مدیریت توان پویا[3] (DPM) به‌عنوان یک استراتژی مؤثر برای بهینه‌سازی مصرف انرژی مطرح است. در این روش، هسته‌های پردازنده در زمان‌هایی که فعالیتی ندارند خاموش می‌شوند و با دریافت وظایف جدید، دوباره فعال می‌گردند. درحالی‌که DVFS به تنظیم هم‌زمان ولتاژ و فرکانس پردازنده می‌پردازد، DPM بر مدیریت خاموش و روشن کردن هسته‌ها تمرکز دارد [5, 13]. تحقیقات اخیر نشان داده‌اند که روش‌های DVFS و DPM ابزارهای مؤثری برای مدیریت توان هستند؛ با این‌حال، توجه بیشتری به بهبود سیاست‌های مدیریت توان برای پردازنده‌های چندهسته‌ای و به‌ویژه بهینه‌سازی استراتژی‌های تصمیم‌گیری برای اعمال DVFS نیاز است. تاکنون توجه کافی به کاهش سربار مدیریت توان و بهبود استراتژی‌های تصمیم‌گیری معطوف نشده است [۵ ,۷-۸,۱۳].

سیاست‌های مدیریت توان به چهار دسته اصلی تقسیم می‌شوند: سیاست‌های زمان‌بندی، پیش‌بینی، تصادفی و یادگیری محور.

سیاست‌های زمان‌بندی تعیین می‌کنند که مؤلفه باید برای مدت‌زمان مشخصی صبر کند تا به حالت کم‌مصرف منتقل شود. سیاست‌های پیش‌بینی براساس پیش‌بینی مدت‌زمان بیکاری آینده تصمیم می‌گیرند که آیا مؤلفه باید به خواب برود یا بیدار بماند. سیاست‌های تصادفی به دودسته زمان‌های گسسته و پیوسته تقسیم می‌شوند و تصمیم‌گیری براساس وقوع رویدادها انجام می‌شود. سیاست‌های یادگیری محور به‌طور پویا با تغییرات بارکاری سازگار می‌شوند و نیازی به دانش پیشین از ماتریس احتمال انتقال حالت ندارند [۱۳].

در همین راستا، روش پیشنهادی از الگوریتم درخت تصمیم‌گیری[4] (DT) برای توسعه مدلی دقیق استفاده می‌کند که قادر به پیش‌بینی پارامترهای کلیدی عملکرد سیستم تحت بارهای کاری مختلف است. سپس، با بهره‌گیری از تکنیک DVFS و الگوریتم رگرسیون خطی، ولتاژ و فرکانس بهینه برای هر هسته تعیین می‌شود. نتایج ارزیابی نشان می‌دهد که مدل پیشنهادی با دقت پیش‌بینی ۹۵ درصد، توانایی پیش‌بینی عملکرد سیستم در شرایط مختلف کاری را دارد و می‌تواند به‌طور مؤثر در طراحی سیستم‌های کم‌مصرف به کار گرفته شود.

فهرست اختصارات و عبارات به‌کاررفته در این مقاله در جدول (۱) ارائه شده است. ساختار مقاله به این شکل است که در بخش دوم، مروری بر کارهای انجام شده‌ی قبلی ارائه می‌شود. بخش سوم به جزئیات چارچوب مدیریت توان مبتنی بر یادگیری تحت نظارت پیشنهادی اختصاص دارد. درنهایت، بخش‌های چهارم و پنجم به نتایج تجربی و نتیجه‌گیری اختصاص‌یافته است.

جدول (۱): فهرست اختصارات و عبارات

ANOVA	Analysis of Variance
DPM	Dynamic Power Management
DT	Decision Tree
DVFS	Dynamic Voltage Frequency Scaling
FN	False Negative
FP	False Positive
FS	Full System
IC	Integrated Circuit
ID3	Iterative Dichotomiser3
KNN	K-Nearest Neighbors
LA- EDF	Look-Ahead Earliest Deadline First
LPM	Local Power Manager
LSTM	Long Short-Term Memory
McPAT	Multicore Power, Area, and Timing
MIPS	Million Instructions Per Second
ML	Machine Learning
MPSoC	Multi-Processor System on a Chip
MSE	Mean Square Error
MSR	Mean Square due to Regression
PCA	Principal Component Analysis
PM	Power Manager
PRE	Precision
RL	Reinforcement Learning
S2S	Seq-to-Seq
TN	True Negative
TP	True Positive
VF	Voltage/Frequency
VLSI	Very-large-scale integration
zTT	Zero Thermal Throttling

۲- کارهای مرتبط

در تلاش برای بهبود DVFS با استفاده از الگوریتم‌های یادگیری ماشین، چندین مطالعه صورت گرفته است. این تحقیقات به موضوعاتی چون کاهش مصرف انرژی، کاهش دما، و افزایش کارایی پرداخته‌اند. برخلاف روش‌های سنتی مدیریت توان که به دانش قبلی از مدل حرارتی و بارهای

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

کاری وابسته‌اند، روش‌های یادگیری ماشین قادرند بدون نیاز به داده‌های پیشین، به‌طور پویا با شرایط متغیر سازگار شوند و از تحلیل رویدادهای گذشته برای بهبود تصمیمات استفاده کنند. در ادامه، به بررسی برخی از این تحقیقات و روش‌های پیشنهادی در زمینه DVFS پرداخته می‌شود.

در مرجع [۷]، یک مدیر توان مبتنی بر یادگیری تحت نظارت برای پردازنده چندهسته‌ای توصیف شده است که از وضعیت عملکرد پردازنده پیش‌بینی شده برای هر کار ورودی برای جستجوی عملکرد بهینه مدیریت توان (ولتاژ - فرکانس) استفاده می‌کند. مقاله [۱۵] رویکردی مبتنی بر یادگیری تقویتی برای مدیریت حرارت در سیستم‌ها ارائه می‌دهد. این روش با یادگیری مداوم و اتخاذ بهترین تصمیم‌ها برای تخصیص وظایف به هسته‌ها و تنظیم پارامترهای سیستم، از بروز مشکلات ناشی از گرمای بیش از حد جلوگیری می‌کند. پژوهش [۱۶]، الگوریتمی برای مدیریت توان و دما مبتنی بر یادگیری تقویتی ارائه می‌دهد که به کنترل تنش‌های دما و توان مصرفی در سیستم‌های چندپردازنده‌ای روی تراشه[۵] (MPSoC) پرداخته است.

کاروالو و همکاران [۲۸] یک مدل غیرخطی مبتنی بر k-نزدیک‌ترین همسایه[۶] (KNN) برای پیش‌بینی فرکانس بهینه پردازنده جهت کاهش مصرف انرژی درگوشی‌های اندرویدی ارائه کرده است. نویسندگان [۲۱] رویکردی برای مدیریت توان ارائه می‌دهند که بر پایه یادگیری تقویتی بدون مدل و طبقه‌بندی بیزی استوار است. این روش با هدف بهینه‌سازی مصرف انرژی و حفظ عملکرد سیستم، به صورت پیوسته و رویداد محور عمل می‌کند. در [۳] رویکرد جدیدی برای مدیریت توان معرفی شده که با استفاده از یادگیری ماشین، ولتاژ و فرکانس پردازنده‌های چندهسته‌ای را تنظیم می‌کند و با پیش‌بینی حجم کاری آینده به بهبود بهینه‌سازی مصرف انرژی کمک می‌کند.

در پژوهشی دیگر [۱۱]، روشی برای مدیریت همکارانه توان پیشنهاد شده است که بر پایه یادگیری تقویتی استوار است. در این روش، دستگاه‌ها تجربیات خود را در زمینه مدیریت انرژی با یکدیگر به اشتراک گذاشته و به مرور زمان بهترین شیوه مدیریت توان را می‌آموزند. در مرجع [۱۷]، یک روش بهبودیافته برای تنظیم پویای ولتاژ و فرکانس بر پایه یادگیری تقویتی ارائه شده است. این روش با هدف کاهش مصرف انرژی در سیستم‌های نهفته چندهسته‌ای، بدون کاهش قابلیت اطمینان وظایف، پارامترهای DVFS را به‌طور دینامیکی تنظیم می‌کند تا تعادل بهینه‌ای بین مصرف انرژی و قابلیت اطمینان ایجاد شود. یک روش بهینه‌سازی توان مبتنی بر الگوریتم Q-learning که به‌طور هم‌زمان محدودیت‌های دما، عملکرد و انرژی را در نظر گرفته و بهترین تنظیمات برای سیستم پیشنهاد شده است [۱۴].

مرجع [۱۸] رویکردی مبتنی بر یادگیری تقویتی برای مدیریت پویای ولتاژ و فرکانس ارائه می‌دهد که باهدف کاهش مصرف انرژی و جلوگیری از کاهش عملکرد ناشی از گرمای بیش‌ازحد طراحی شده است. نویسندگان [۲۵-۲۴, ۱۰] الگوریتمی را معرفی کرده‌اند که با الهام از روش سنتی LA-EDF[۷] طراحی شده [۲۶] و با ترکیب یادگیری تقویتی، به بهبود بهینه‌سازی مصرف انرژی در سیستم‌های بلادرنگ پرداخته است. نویسندگان [۱۲,۲۳] تکنیک مبتنی بر یادگیری عمیق را برای بهینه‌سازی مصرف انرژی در پردازنده‌های تک‌هسته‌ای ارائه می‌دهند. این روش از مدل رمزگشا-رمزگذار مبتنی بر شبکه‌های LSTM[۸] استفاده نموده و S2S-LSTM[۹] استفاده می‌کند تا DVFS را به‌عنوان یک مسئله شبکه بازگشتی مدل‌سازی کند. در [۱۰]، روشی برای کاهش مصرف انرژی در سیستم‌های محاسباتی ارائه شده است که با بهره‌گیری از الگوریتم یادگیری تقویتی، ولتاژ و فرکانس پردازنده را متناسب با تغییرات بارکاری به طور خودکار تنظیم می‌کند.

جدول (۲) به‌طور خلاصه عملکرد این روش‌ها را در زمینه DVFS نمایش می‌دهد.

۳- روش پیشنهادی

در مطالعه حاضر، یک روش مدیریت توان با استفاده از تکنیک‌های یادگیری ماشین برای پیش‌بینی عملکرد سیستم ارائه شده و سپس سطح ولتاژ و فرکانس متناسب با عملکرد پیش‌بینی شده سیستم به بار کاری اختصاص داده می‌شود. این روش ازجمله روش‌های سطح بالا و انتزاعی است که از عملکرد سیستم در برابر خرابی‌های زمان اجرا به دلیل تغییرات زمانی محافظت می‌کند و پارامترهای سیستم را به‌صورت پویا تنظیم می‌نماید. همچنین، ازنظر بهبود طول عمر، می‌تواند کارآمدتر باشد. رویکرد ارائه شده باتوجه به شکل (۱) به شرح زیر است که هرکدام در ادامه توضیح مختصر داده شده‌اند:

- استخراج ویژگی‌های ورودی و معیارهای خروجی
- طبقه‌بندی و پیش‌بینی وضعیت سیستم
- تخصیص ولتاژ و فرکانس به بار کاری

۳-۱- استخراج ویژگی‌های ورودی و معیارهای خروجی

در فاز اول، داده‌های موردنیاز از یک سیستم پردازنده چندهسته‌ای استخراج شده است که از تکنیک DVFS برای تنظیم سطوح مختلف ولتاژ و فرکانس جهت اجرای بارهای کاری بهره می‌برد، زیرا مجموعه‌داده‌ها نقش کلیدی در تحقیقات یادگیری ماشین دارند و برای ارزیابی الگوریتم‌ها و بهبود عملکرد سیستم حیاتی هستند. برای انتخاب ویژگی‌های ورودی مؤثر، از رگرسیون لاسو به‌عنوان یک تکنیک کاهش ابعاد استفاده شده است. لاسو با کمینه‌سازی هم‌زمان مجموع مربعات خطا و مجموع قدرمطلق ضرایب، به‌طور خودکار ویژگی‌های کم‌اهمیت را حذف کرده و تعداد پارامترهای مدل را کنترل می‌کند. این فرایند، مدل را ساده‌تر کرده و ضمن بهبود تعمیم‌پذیری، خطر بیش برازش را کاهش می‌دهد [۲۸]. در این پژوهش، عملکرد سیستم بر اساس معیارهای توان مصرفی، انرژی مصرفی، زمان اجرا و عملکرد محاسباتی ارزیابی شده است. زمان اجرا به‌عنوان معیار اصلی انتخاب شده، زیرا یکی از کامل‌ترین و قابل‌اعتمادترین سنجش‌های عملکرد سیستم است.

[Downloaded from jiaeee.com on 2026-11-16]

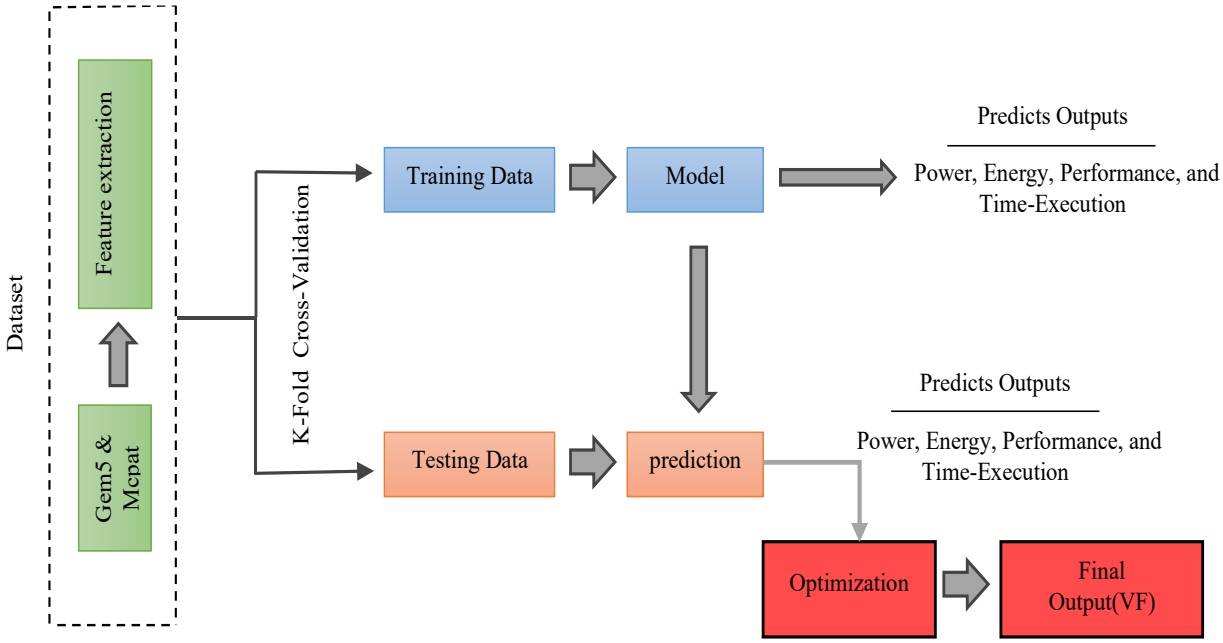
[DOI: 10.61186/jiaeee.22.1.99]

جدول (۲): خلاصه کارهای پیشین

سال	شماره مرجع	هدف بهینه سازی	تکنیک یادگیری	مزایا	معایب
۲۰۱۰	[۷]	توان/انرژی	یادگیری با ناظر	کاهش مصرف توان بدون افت عملکرد، سازگاری با تغییرات بار کاری	نیاز به یک جدول سیاست پیش‌فرض، نیاز به داده‌های آموزشی و پیچیدگی پیاده‌سازی
۲۰۱۴	[۱۵]	قابلیت اطمینان (طول عمر)	یادگیری تقویتی	بهبود طول عمر سیستم، سازگاری با تغییرات دینامیکی محیطی	پیچیدگی پیاده‌سازی، زمان آموزش طولانی، مدل دقیق از سیستم برای آموزش الگوریتم
۲۰۱۵	[۱۶]	توان/انرژی، دما	یادگیری تقویتی	انعطاف پذیری، کاهش توان مصرفی. کنترل دمای و حفظ عملکرد	وابستگی به دقت مدل، نیاز به داده‌های آموزشی و پیچیدگی پیاده‌سازی
۲۰۱۶	[۲۸]	توان/انرژی	یادگیری با ناظر	کاهش مصرف انرژی، دقت بالای مدل	زمان محاسباتی بالا، کاهش دقت مدل در صورت افزایش تعداد همسایگان
۲۰۱۶	[۲۱]	توان/انرژی، کارایی	یادگیری تقویتی، یادگیری با ناظر	انعطاف پذیری، کاهش توان مصرفی و حفظ عملکرد، پیش بینی حجم کار	وابستگی به دقت مدل، نیاز به داده‌های آموزشی و پیچیدگی پیاده‌سازی
۲۰۱۸	[۳]	توان/انرژی	یادگیری با ناظر	انعطاف پذیری، کاهش توان مصرفی و حفظ عملکرد، یادگیری مداوم	پیچیدگی پیاده‌سازی، وابستگی مستقیم دقت مدل یادگیری ماشین به کیفیت داده‌های آموزشی
۲۰۱۸	[۱۱]	توان/انرژی	یادگیری تقویتی	سرعت بالای فرایند یادگیری، بهبود کیفیت سیاست‌های یادگرفته شد، کاهش مصرف انرژی	پیچیدگی پیاده‌سازی، نیاز به ارتباطات شبکه‌ای، احتمال سربار کنترلی
۲۰۲۰	[۱۷]	توان/انرژی، قابلیت اطمینان	یادگیری تقویتی	کاهش توان مصرفی، حفظ قابلیت اطمینان وظایف، سازگاری با شرایط مختلف سیستم	پیچیدگی پیاده‌سازی، زمان آموزش طولانی،
۲۰۲۰	[۲۴]	توان/انرژی	یادگیری تقویتی	در نظر گرفتن محدودیت‌های چندگانه (دما، عملکرد)، انطباق با شرایط مختلف، کاهش مصرف انرژی	پیاده‌سازی پیچیده، زمان آموزش طولانی مدل، وابستگی به دقت مدل و نیاز به تنظیم دقیق پارامترها
۲۰۲۱	[۱۶]	توان/انرژی	یادگیری تقویتی	بهینه‌سازی مصرف انرژی، انعطاف‌پذیری،	پیچیدگی پیاده سازی، پیچیدگی پیاده‌سازی، زمان آموزش طولانی
۲۰۲۱	[۲۳]	توان/انرژی	یادگیری تقویتی	کاهش مصرف انرژی، بهبود عملکرد، تطبیق‌پذیری با تغییرات بار کاری و محیط سیستم	پیچیدگی محاسباتی، نیاز به داده‌های آموزشی، وابستگی به کیفیت مدل یادگیری تقویتی
۲۰۲۳	[۱۲]	توان/انرژی	یادگیری با ناظر	بهینه سازی مصرف انرژی، حفظ عملکرد سیستم	پیچیدگی پیاده‌سازی، نیازمند به اصلاح و تنظیمات برای پردازنده های چند هسته ای
۲۰۲۴	[۱۰]	توان/انرژی	یادگیری تقویتی	انعطاف پذیری، کاهش مصرف انرژی، بهبود عملکرد	پیچیدگی پیاده‌سازی، وابستگی به دقت مدل و نیاز به تنظیم دقیق پارامترها، چالش‌های زمان واقعی

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]



شکل (۱): رویکرد کلی برای تخصیص جفت VF به بارکاری

همچنین، شاخص میلیون دستورالعمل در ثانیه[۱۰] (MIPS) برای ارزیابی کارایی سیستم استفاده شده که نرخ اجرای دستورالعمل‌ها را اندازه‌گیری می‌کند و رابطه‌ای معکوس بازمان اجرا دارد، بنابراین برای توصیف کارایی و سرعت سیستم مناسب است [۲۹]. ویژگی‌های خروجی باکلاس‌های مجموعه L برچسب‌گذاری می‌شود که در جدول (۳) تعریف شده و هر کلاس نشان‌دهنده یک سطح یا محدوده از پیش تعیین شده، نظیر محدوده توان مصرفی یا زمان اجرای کار است. این محدوده‌ها در جدول (۴) ارائه شده‌اند. هدف این مرحله، شناسایی ویژگی‌های مؤثر بر عملکرد سیستم است.

۳-۲- طبقه‌بندی و پیش‌بینی وضعیت سیستم

پس از گردآوری داده‌ها و آماده‌سازی مجموعه آموزشی، فاز دوم شامل طبقه‌بندی با استفاده از یادگیری نظارت شده است که هدف آن پیش‌بینی برچسب کلاس ویژگی‌های خروجی بر اساس ورودی‌ها است. این روش برای کشف دانش و استخراج استراتژی‌های تصمیم‌گیری خود بهبود بخش مفید است. برای بهبود دقت، لازم است بازه ویژگی‌های ورودی و خروجی به یکدیگر نزدیک باشد. بدین منظور از روش نرمال‌سازی Min-Max استفاده می‌شود که داده‌ها را در بازه (۰،۱) مقیاس‌بندی می‌کند. این تبدیل با استفاده از رابطه (۱) انجام می‌شود:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (۱)$$

در این رابطه، X_{min} حداقل مقدار و X_{max} حداکثر مقدار را نشان می‌دهند. همچنین، با استفاده از Grid-Search و K-Fold Cross-Validation، هایپرپارامترها بهینه‌سازی شده و عملکرد مدل با تقسیم داده‌ها به دسته‌های آموزش و آزمون ارزیابی می‌شود. این روش‌ها به کاهش تأثیر تقسیم داده‌ها بر نتایج و ارائه تخمینی دقیق از عملکرد مدل کمک می‌کنند. در این مطالعه، الگوریتم درخت تصمیم به دلیل توانایی بالای آن در دسته‌بندی داده‌ها و پیش‌بینی دقیق حالت‌های مختلف عملکرد سیستم انتخاب شد. این الگوریتم به‌خاطر ساختار ساده و قابل‌فهم، امکان تحلیل و طبقه‌بندی نمونه‌ها را براساس ویژگی‌های ورودی فراهم می‌کند و با داده‌های گسسته و پیوسته به‌خوبی سازگار است.

برخلاف بسیاری از الگوریتم‌های دیگر، DT قادر به حل هم‌زمان مسائل رگرسیون و طبقه‌بندی است، هرچند بیشتر برای طبقه‌بندی استفاده می‌شود. مزایای کلیدی آن شامل سادگی در تفسیر، سرعت در یادگیری و عدم نیاز به پیش‌فرض‌های پیچیده است. همچنین، توانایی این الگوریتم در انتخاب ویژگی‌های کلیدی در هر مرحله از تصمیم‌گیری، آن را به یک ابزار دقیق برای پیش‌بینی عملکرد سیستم تبدیل کرده است. الگوریتم DT با شروع از ریشه، نمونه‌ها را از طریق یک سری از تصمیمات متوالی به دسته‌بندی‌های مختلف هدایت می‌کند. در هر مرحله، الگوریتم یک ویژگی از داده‌ها را انتخاب می‌کند که بیشترین توانایی برای تفکیک داده‌ها به کلاس‌های مختلف را دارد [۳۲]. در این پروژه، برای انتخاب ویژگی در هر گره از درخت، از شاخص جینی استفاده شده است که میزان نابرابری در توزیع کلاس‌ها را محاسبه می‌کند. شاخص جینی با استفاده از رابطه (۲) تعریف می‌شود [۳۲]:

[Downloaded from jiaeee.com on 2026-11-16]
[DOI: 10.61186/jiaeee.22.1.99]

جدول (۳): تعریف کلاس‌های خروجی

L1={power1,power2,power3}	power1> power2> power3
L2={Energy1, Energy2, Energy3}	Energy1> Energy2> Energy2
L3={performance_Time1, performance_Time2, performance_Time3 }	performance_Time1> performance_Time2> performance_Time3
L4={Mips1,mips2,mips3}	Mips1> Mips2> Mips2
L5={ Execution_Time1, Execution_Time2, Execution_Time3}	Execution_Time1> Execution_Time2> Execution_Time3

جدول (۴): نمایش بازه بندی هر کلاس از ویژگی هدف

ویژگی‌های خروجی	مقادیر هر بازه		
power	power1: (1, 0.000179)	power2: (0.000177, 0.00012)	power3: (0.00012)
energy	Energy1: (1, 0.000046)	Energy2: (0.000046, 0.000035)	Energy3: (0,0.00003)
Performance (Execution Time)	Performance_ Time1: (1,0.0023)	Performance_Time2: (0.002338,0.0018)	Performance_Time3: (0,0.0018)
Performance (MIPS)	Mips1: (1, 0.64)	Mips2: (0.64, 0.27)	Mips3: (0,0.26)
Execution Time	Execution_Time1: (1,0.00033)	Execution_Time2: (0.00033, 0.00025)	Execution_Time3: (0,0.0024)

جدول (۵): ضرایب رگرسیون برای متغیرهای مستقل و وابسته

Indepnd / Dependent		power	energy	Performance (Execution Time)	Performance (MIPS)	Execution Time
Voltage	0.74	0.092	0.040	0.90-	0.089-	0.25
Frequency	1.24	-0.12	-0.037	0.054	-0.04	-0.04

$$Gini = 1 - \sum_{i=1}^{n} {p_i}^2 \qquad (۲)$$

که در آن، p احتمال قرارگرفتن یک نمونه در یک کلاس خاص است. ویژگی‌ای که کمترین مقدار شاخص جینی را در هر گره داشته باشد، به‌عنوان مبنای تقسیم‌بندی داده‌ها در آن گره انتخاب می‌شود. برای داده‌های پیوسته، ویژگی‌های عددی با بازه‌بندی به داده‌های گسسته تبدیل می‌شوند. به عنوان مثال، یکی از ویژگی‌های کلیدی در این مطالعه Power بود که با استفاده از شاخص جینی، مقدار آستانه برای آن در حدود ۴،۷۵- تعیین شد. مقادیر کمتر یا مساوی ۴،۷۵- به یک شاخه از درخت و مقادیر بیشتر از آن به شاخه دیگر هدایت شدند. این فرآیند برای سایر ویژگی‌ها نیز ادامه می‌یابد تا نمونه‌ها به دسته‌های نهایی خود برسند. این روش این امکان را فراهم می‌کند که برای ورودی‌های جدید، از ریشه درخت شروع کرده و بر اساس ویژگی‌های هر گره، تا رسیدن به برگ‌های نهایی پیش رود و به دسته‌بندی نهایی دست یابد.

۳-۳- تخصیص ولتاژ و فرکانس به بارکاری

در مرحله نهایی تخصیص ولتاژ و فرکانس، به‌منظور بهینه‌سازی عملکرد سیستم، از رگرسیون خطی استفاده شده است. دلیل انتخاب این روش سادگی مفهوم آن، کارایی بالا و نیاز کمتر به محاسبات پیچیده در مقایسه با دیگر روش‌های تخمین است. رگرسیون خطی به‌خوبی برای مدل‌سازی روابط خطی میان متغیرهای وابسته و مستقل مناسب است و به‌ویژه در مسائلی که روابط میان داده‌ها ساده و خطی باشند، عملکرد بسیار مؤثری دارد. رگرسیون خطی به‌عنوان یک روش با ناظر در یادگیری ماشین، برای مدل‌سازی روابط بین متغیرهای مستقل (توان مصرفی، انرژی، زمان اجرا، و کارایی) و متغیر وابسته (ولتاژ و فرکانس) استفاده می‌شود. هدف از این روش، حداقل کردن مجموع مربعات خطا یا همان تابع هزینه بین مشاهدات واقعی و مقادیر پیش‌بینی شده است که باعث بهبود دقت مدل در پیش‌بینی ولتاژ و فرکانس بهینه می‌شود.

فرمول شاخص خطا در رگرسیون خطی با استفاده از رابطه (۳) تعریف می‌شود [۳۲]:

$$J(\theta_0, \theta_1) = \frac{1}{n}\sum_{i=1}^{n}(h_\theta(x_i) - y_i)^2 \qquad (۳)$$

در این رابطه، $h_\theta(x_i)$ مقادیر پیش‌بینی شده و y_i مقادیر واقعی متغیر وابسته است. با داشتن بیش از یک متغیر مستقل، رگرسیون به

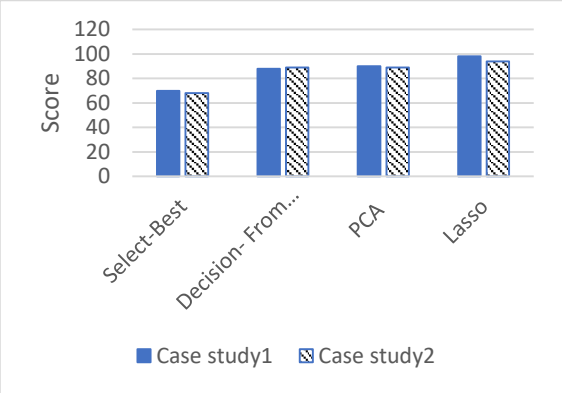
[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

شکل چندمتغیره در نظر گرفته می‌شود و رابطه بین متغیرهای مستقل و وابسته با استفاده از رابطه (۴) مدل‌سازی می‌شود [۳۲].

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \quad (۴)$$

در این معادله، β_0 به‌عنوان عرض از مبدأ و β_1 تا β_n شیب‌های خط هستند. شیب‌خط نشان‌دهنده میزان حساسیت متغیر وابسته نسبت به متغیرهای مستقل است. عرض از مبدأ نیز مقدار متغیر وابسته را به ازای صفر بودن متغیرهای مستقل نمایش می‌دهد. ضرایب رگرسیون به شکلی محاسبه می‌شوند که مجموع مربعات خطا بین مشاهدات و خروجی مدل به حداقل برسد. این ضرایب با صفر کردن مشتق تابع هزینه به دست می‌آیند. ضرایب محاسبه شده برای متغیرهای مستقل (توان مصرفی، انرژی، زمان اجرا، و کارایی) و متغیر وابسته (ولتاژ و فرکانس) در جدول (۵) آمده‌اند که نتایج به‌دست‌آمده از این محاسبات را نشان می‌دهند.



شکل (۲): مقایسه روش‌های کاهش ابعاد

۴- آزمایش‌ها و ارزیابی نتایج

۴-۱- تنظیمات آزمایش

برای پیاده‌سازی و ارزیابی روش پیشنهادی از شبیه‌ساز Gem5 در حالت سیستم کامل[۱۱](FS) همراه با ابزار McPAT[۱۲] استفاده شده است. McPAT برای محاسبه پارامترهایی نظیر توان مصرفی، انرژی و کارایی محاسباتی به کار گرفته می‌شود. شبیه‌ساز Gem5 به دلیل توانایی شبیه‌سازی سیستم‌عامل و مدیریت وقفه‌ها، استثنائات، و سطوح دسترسی، به‌عنوان یکی از بهترین گزینه‌ها برای ارزیابی سیستم‌های سطح سیستم انتخاب شده است. در این شبیه‌سازی، یک پردازنده چهار هسته‌ای همگن باقابلیت تنظیم فرکانس هر هسته بین بازه‌ای از f_{min} تا f_{max} و ولتاژ بین v_{min} تا v_{max} استفاده شده است.

به دلیل تفاوت در بارهای کاری، فرکانس هسته‌ها متفاوت است تا عملکرد بهینه تضمیشود. هسته‌هایی که در سطوح ولتاژ و فرکانس بالاتر فعالیت می‌کنند، انرژی بیشتری مصرف کرده و کارایی بهتری دارند. معماری مورداستفاده شامل حافظه نهان دوسطحی است: سطح اول برای داده‌ها و دستورالعمل‌ها و سطح دوم به‌صورت اشتراکی بین هسته‌ها. از معماری ARM Cortex-A باقابلیت تنظیم دینامیک ولتاژ و فرکانس استفاده شده است. برای ارزیابی مدل پیشنهادی، از مجموعه برنامه‌های بسته محک SPEC-CPU-2006 به‌عنوان بارکاری استفاده شده تا داده‌هایی برای مدیریت توان پویا با استفاده از الگوریتم‌های یادگیری ماشین تولید شود. جزئیات کامل تنظیمات شبیه‌سازی سیستم در جدول (۶) ارائه شده است.

۴-۲- نتایج و بحث

با اجرای مجموعه‌ای از وظایف روی پردازنده چندهسته‌ای، دو مجموعه‌داده برای مطالعه موردی۱ و مطالعه موردی۲ ایجاد شد. به دلیل بیشتر بودن تعداد ویژگی‌ها نسبت به نمونه‌ها، کاهش ابعاد ضروری بود تا دقت طبقه‌بندی بهبود یابد و زمان اجرای مدل‌ها کاهش پیدا کند. برای این منظور، چهار روش کاهش ابعاد شامل Lasso، PCA[۱۳]، Select-KBest و Select-From-Model ارزیابی شدند. نتایج نشان داد که روش Lasso نسبت به سایر روش‌ها عملکرد بهتری داشت و دقت بالاتری در پیش‌بینی‌ها ارائه داد. بنابراین، این روش به دلیل بهبود عملکرد مدل و کاهش مؤثر ابعاد، به‌عنوان روش اصلی کاهش ابعاد در این پژوهش انتخاب شد و در پیش‌بینی عملکرد سیستم تحت بارهای کاری مختلف مورداستفاده قرار گرفت. نتایج مقایسه در شکل (۲) ارائه شده است.

برای اعمال روش پیشنهادی، از الگوریتم درخت تصمیم استفاده شد که به دلیل سادگی و قابلیت تفسیر بالا. کارایی بالا در طبقه‌بندی داده‌ها، به‌عنوان یکی از روش‌های اصلی طبقه‌بندی انتخاب گردید. علاوه بر این، از دو الگوریتم دیگر، یعنی الگوریتم بیزی[۱۴] [7] و الگوریتم K-NN [28]، نیز برای طبقه‌بندی داده‌ها و ارزیابی مدل‌ها بهره بردیم. ارزیابی مدل‌ها با استفاده از معیارهای دقت[۱۵]، صحت (precision)، فراخوانی (recall)، و امتیاز F1 (میانگین هارمونیک دقت و صحت) انجام شد.

دقت: نسبت پیش‌بینی‌هایی که مدل به‌درستی طبقه‌بندی کرده است. این معیار متداول‌ترین و اساسی‌ترین سنجش کیفیت الگوریتم‌های طبقه‌بندی است.

$$\text{Accuracy} = \frac{^{۱۶}\text{TP} + ^{۱۷}\text{TN}}{\text{TP} + \text{TN} + ^{۱۸}\text{FP} + ^{۱۹}\text{FN}} \quad (۵)$$

- **صحت:** نسبت نمونه‌های مثبت صحیح طبقه‌بندی شده به‌کل نمونه‌های طبقه‌بندی شده مثبت. این معیار نشان می‌دهد که از بین تمام نمونه‌هایی که مدل به‌عنوان مثبت شناسایی کرده است، چه تعداد واقعاً به کلاس مثبت تعلق دارند.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (۶)$$

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

جدول (۶): جزئیات پیکربندی سیستم

پارامترها	تنظیمات
Core- Type	ARM Cortex-A7
Core Microarchitecture	ARMv7-A
The Number of Cores	4-core system
Core Voltage	(1.278V,0.926V)
Core Frequency	(1.7GHZ,0.2GHZ)
Memory	mem-type:DDR3_1600_8x8, mem-size:256MB
L1-I Cache	32kB
L1-D Cache	32kB
L2 Cache	256kB

- **فراخوانی:** فراخوانی که به‌عنوان حساسیت یا نرخ مثبت واقعی نیز شناخته می‌شود، درصد نمونه‌های مثبت واقعی است که به‌درستی شناسایی شده‌اند و قدرت شناسایی مدل را نشان می‌دهد.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (۷)$$

- **امتیازF1** : میانگین هارمونیک دقت و فراخوانی که به‌طورکلی کارایی مدل را در تعادل بین دقت و شناسایی مثبت‌ها ارزیابی می‌کند.

$$\text{F1} - \text{Score} = \frac{\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (۸)$$

در روابط فوق منظور از TF،TP ، FP و FN به شرح زیر است:

- TP: مثبت واقعی که به‌درستی پیش‌بینی کند.
- TN: منفی واقعی که مدل پیش‌بینی می‌کند.
- FP: مثبت کاذب، یا خطای نوع ۱، که در آن مدل به اشتباه کلاس مثبت را پیش‌بینی کرده است.
- FN: منفی کاذب، یا خطای نوع ۲ که در آن مدل به طور نادرست کلاس منفی را پیش‌بینی کرده است.

نتایج مقایسه مدل‌ها در شکل‌های ۳ تا ۶ نمایش داده شده است و تحلیل و تفسیر این نتایج بر اساس نمودارها به شرح زیر است:

- الگوریتم درخت تصمیم بالاترین دقت را در مقایسه با سایر الگوریتم‌ها به دست آورده است. این الگوریتم دارای بیشترین مقادیر TP و TN است و در پیش‌بینی عملکرد سیستم در تمام پنج پارامتر موردنظر به‌خوبی عمل کرده است.
- الگوریتم بیزی نسبت به سایر الگوریتم‌ها عملکرد ضعیف‌تری داشته و کمترین دقت را در هر دو مطالعه موردی ۱ و مطالعه موردی ۲ ارائه داده است.
- استفاده از الگوریتم‌ها در مطالعه موردی ۱ به‌طورکلی عملکرد بهتری نسبت به مطالعه موردی ۲ داشته است.

با توجه به اینکه پیچیدگی زمانی یک تکنیک، نقش مهمی در عملکرد آن دارد و می‌تواند به‌صورت تجربی از زمان اجرای آن مشهود باشد، می‌توان الگوریتم‌ها را بر اساس میزان کارایی‌شان ارزیابی کرد. در تحلیل پیچیدگی زمانی، الگوریتم k-نزدیک‌ترین همسایه (K-NN) با مرتبه زمانی O(n2)به دلیل نیاز به محاسبه فاصله بین تمام داده‌ها، زمان اجرای بالاتری نسبت به درخت تصمیم با مرتبه زمانی O(nlogn) دارد. این مسئله باعث می‌شود که k-NN در مواجهه با مجموعه‌داده‌های بزرگ عملکرد ضعیف‌تری داشته باشد. از سوی دیگر، الگوریتم بیزی با مرتبه زمانی خطی O(n) به لحاظ پیچیدگی زمانی بسیار کارآمدتر است، اما دقت کمتری نسبت به درخت تصمیم دارد. این کاهش دقت به دلیل ساده‌سازی‌هایی است که در مدل بیزی انجام می‌شود. درخت تصمیم، با وجود پیچیدگی زمانی بیشتر نسبت به بیزی، به دلیل ساختار سلسله مراتبیش توانایی بهتری در تفکیک داده‌ها دارد و به همین دلیل تعادل بهتری بین دقت و پیچیدگی زمانی ارائه می‌دهد [32].

برای ارزیابی عملکرد الگوریتم پس از پیش‌بینی حالت‌های عملکرد سیستم و تخصیص جفت ولتاژ و فرکانس، از تحلیل واریانس ANOVA استفاده شد. این تحلیل بر اساس تقسیم واریانس متغیرهای وابسته به

جدول (۷) : نتایج تحلیل رگرسیون برای پیش‌بینی فرکانس

Model	Sum of Squarea	df	Mean Square	F	R_square	Adjusted_R_square
Regression	1.65	5	0.33	9.66	0.17	0.15
Residual	8.15	218	0.034			
Total	9.8	223				
Dependent Variable: freuence						
Predictors (Constant), power, energy, Execution_Time, performance_Time, performance_mips						

[Downloaded from jiaeee.com on 2026-11-16]

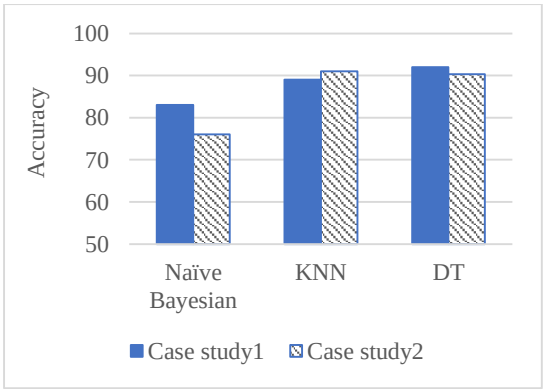
[DOI: 10.61186/jiaeee.22.1.99]

مجموع مربعات خطا[۲۰] (SSE): برای بررسی دقت مدل، تفاوت بین مقادیر واقعی و پیش‌بینی شده محاسبه شده است. مقادیر ۰،۰۳۳ و ۰،۰۰۵ برای متغیرهای وابسته نشان می‌دهد که خطا بسیار کم و نزدیک به صفر است، که بیانگر عملکرد مطلوب مدل است.

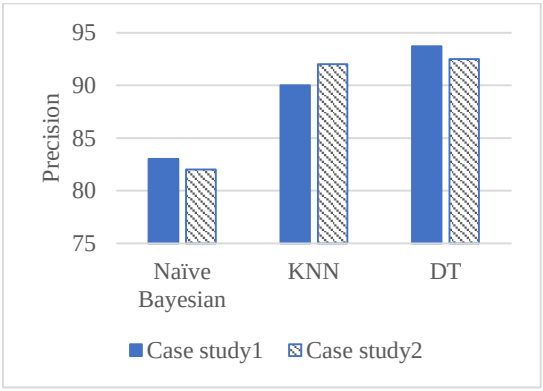
- **ضریب تعیین(R^2)** : مقادیر ۰،۱۷ و ۰،۰۸ نشان‌دهنده‌ی آن است که مدل به ترتیب ۲۰٪ و ۱۰٪ از پراکندگی متغیرهای وابسته را توضیح می‌دهد. هرچه این مقادیر به ۱ نزدیک‌تر باشد، مدل بهتر می‌تواند تغییرات متغیر وابسته را توجیه کند.
- **ضریب تعیین اصلاح شده:** مقادیر ضریب تعیین اصلاح شده برابر با ۰،۱۵ و ۰،۰۶ به‌دست‌آمده است. نزدیکی این مقادیر به R^2 نشان می‌دهد که متغیرهای مستقل در مدل به‌خوبی توانسته‌اند تغییرات متغیر وابسته را توضیح دهند.
- **نسبت F:** نسبت F که از تقسیم MSR به MSE محاسبه می‌شود، مقادیر ۹،۶ و ۴،۰۵ را نشان می‌دهد. بزرگی این مقادیر نشان‌دهنده آن است که مدل رگرسیون به‌طور مؤثری پراکندگی کل را نسبت به باقی‌مانده‌ها توضیح می‌دهد و بنابراین مدل مناسبی است.

۵- نتیجه‌گیری

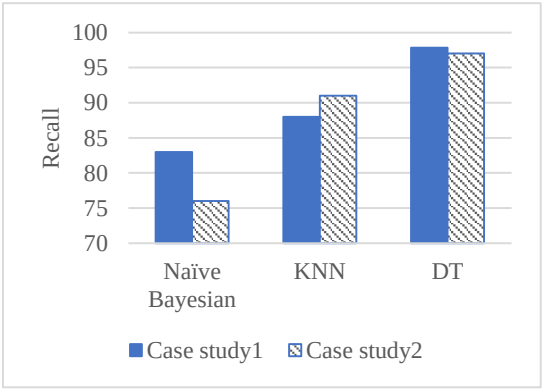
در این مطالعه، رویکردی برای مدیریت توان پردازنده‌های چندهسته‌ای با استفاده از تکنیک مقیاس‌بندی پویا ولتاژ و فرکانس و یادگیری ماشین معرفی شد. این روش با بهره‌گیری از الگوریتم‌های طبقه‌بندی، توانسته است سطوح بهینه ولتاژ و فرکانس را به‌طور دقیق برای بارهای کاری مختلف پیش‌بینی کند. به‌کارگیری این رویکرد منجر به کاهش سربارهای پردازشی و بهبود کارایی کلی سیستم گردید. نتایج تجربی نشان داد که این روش به‌خوبی با تغییرات سیستم و حجم‌های کاری مختلف سازگار است و با بهینه‌سازی تصمیمات مدیریتی بر اساس داده‌های گذشته، به بهبود عملکرد و افزایش طول عمر سیستم کمک می‌کند.



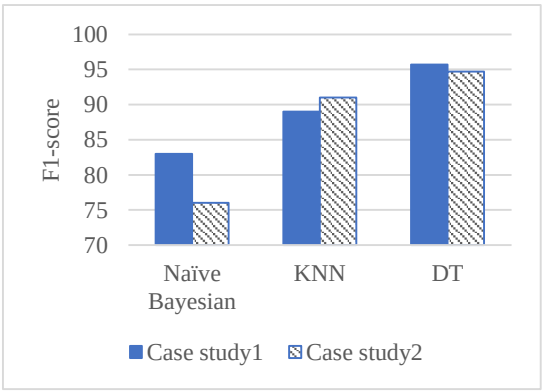
شکل (۳): مقایسه براساس معیار دقت



شکل (۴): مقایسه براساس معیار صحت



شکل (۵): مقایسه براساس معیار فراخوانی



شکل (۶) مقایسه براساس معیار امتیاز F1

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

جدول (۸): نتایج تحلیل رگرسیون برای پیش‌بینی ولتاژ

Model	Sum of Squarea	df	Mean Square	F	R_square	Adjusted_R_square
Regression	0.111	5	0.02	4.05	0.08	0.06
Residual	1.31	218	0.005			
Total	1.42	223				
Dependent Variable: voltage						
Predictors (Constant), power, energy, Execution_Time, performance_Time, performance_mips						

مراجع

[1] Sharafinejad R, Alizadeh B. Formal Verification of System-Level Power Management Architecture in Modern Processors. Journal of Iranian Association of Electrical and Electronics Engineers 2021; 18 (4) :185-196

[2] Nori M, Bekrani M. Quasi-floating gate MOSFET-based low power current mirror. Journal of Iranian Association of Electrical and Electronics Engineers 2024; 21 (1) :27-38

[3] P. Manoj, A. Jantsch, and M. Shafique, "SmartDPM: Machine Learning-Based Dynamic Power Management for Multi-Core Microprocessors", Journal of Low Power Electronics, vol. 14, no. 4, pp. 460-474, 2018.

[4] L. R. Juracy, M. T. Moreira, A. d. M. Amory, and F. G. Moraes, "A survey of aging monitors and reconfiguration techniques", arXiv preprint arXiv:2007.07829, 2020.

[5] S. Pagani, P. S. Manoj, A. Jantsch, and J. Henkel, "Machine learning for power, energy, and thermal management on multicore processors: A survey", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 1, pp. 101-116, 2018.

[6] F. A. Endo, "Génération dynamique de code pour l'optimisation énergétique", Université Grenoble Alpes (ComUE), 2015.

[7] H. Jung and M. Pedram, "Supervised learning based power management for multicore processors", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 9, pp. 1395-1408, 2010.

[8] H. Shen, J. Lu, and Q. Qiu, "Learning based DVFS for simultaneous temperature, performance and energy management", in Thirteenth International Symposium on Quality Electronic Design (ISQED), 2012, pp. 747-754: IEEE.

[9] Panda, Prasanta, Aruna Tripathy, and Kanhu Charan Bhuyan. "Reinforcement Learning-Based Dynamic Voltage and Frequency Scaling for Energy-Efficient Computing", 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). IEEE, 2024.

[10] K. Yu, "Deep Reinforcement Learning Based DVFS Algorithm Frameworks", SWINBURNE UNIVERSITY OF TECHNOLOGY, 2022.

[11] Tian, Zhongyuan, et al. "Collaborative power management through knowledge sharing among multiple devices", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38.7 (2018): 1203-1215.

[12] S. K. Thethi and R. Kumar, "Power optimization of a single-core processor using LSTM based encoder–decoder model for online DVFS", Sādhanā, vol. 48, no. 2, p. 37, 2023.

[13] S. Yue, D. Zhu, Y. Wang, and M. Pedram, "Reinforcement learning based dynamic power management with a hybrid power supply", in 2012 IEEE 30th International Conference on Computer Design (ICCD), 2012, pp. 81-86: IEEE.

[14] L. Li et al., "An Improved Q-Learning for System Power Optimization with Temperature, Performance and Energy Constraint Modeling", in 2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS).

[15] A. Das, R. A. Shafik, G. V. Merrett, B. M. Al-Hashimi, A. Kumar, and B. Veeravalli, "Reinforcement learning-based inter-and intra-application thermal optimization for lifetime improvement of multicore systems", in Proceedings of the 51st Annual Design Automation Conference, 2014, pp. 1-6.

[16] A. Iranfar, S. N. Shahsavani, M. Kamal, and A. Afzali-Kusha, "A heuristic machine learning-based algorithm for power and thermal management of heterogeneous MPSoCs", in 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), 2015, pp. 291-296: IEEE.

[17] A. Yeganeh-Khaksar, M. Ansari, S. Safari, S. Yari-Karin, and A. Ejlali, "Ring-DVFS: Reliability-aware reinforcement learning-based DVFS for real-time embedded systems", IEEE Embedded Systems Letters, vol. 13, no. 3, pp. 146-149, 2020.

[18] S. Kim, K. Bin, S. Ha, K. Lee, and S. Chong, "zTT: Learning-based DVFs with zero thermal throttling for mobile devices", in Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, 2021, pp. 41-53.

[19] C. Robert, "Machine learning, a probabilistic perspective", ed: Taylor & Francis, 2014.

[20] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, Mathematics for machine learning. Cambridge University Press, 2020.

[21] Y. Wang and M. Pedram, "Model-free reinforcement learning and bayesian classification in system-level power management", IEEE Transactions on Computers, vol. 65, pp. 3713-3726, 2016.

[22] V. Spiliopoulos, A. Bagdia, A. Hansson, P. Aldworth, and S. Kaxiras, "Introducing DVFS-management in a full-system simulator", in 2013 IEEE 21st International symposium on modelling, analysis and simulation of computer and telecommunication systems, 2013, pp. 535-545: IEEE.

[Downloaded from jiaeee.com on 2026-11-16]

[DOI: 10.61186/jiaeee.22.1.99]

[23] S. K. Thethi and R. Kumar, “Online DVFS using Deep Learning: Sequence to Sequence LSTM Networks with Attention”, 2022.
[24] Yu, Ke, et al. “An improved DVFS algorithm for energy-efficient real-time task scheduling”, 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2020.
[25] Yu, Ke, et al. “A Framework for Deep Q-Learning Based Hybrid DVFS Algorithms for Real-Time Systems”, 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, 2021.
[26] P. Pillai and K. G. Shin, “Real-time dynamic voltage scaling for low-power embedded operating systems”, in Proceedings of the eighteenth ACM symposium on Operating systems principles, 2001, pp. 89-102.
[27] Farzane Nakhaee et al. 2017. Lifetime improvement by exploiting aggressive voltage scaling during runtime of error-resilient applications. Integration, the VLSI Journal (2017).
[28] Carvalho, Sidartha AL, Daniel C. Cunha, and Abel G. Silva-Filho. “On the use of nonlinear methods for low-power CPU frequency prediction based on Android context variables”, 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA). IEEE, 2016.
[29] Patterson, David A. “Computer Organization and Design”, (2022).
[30] C. Robert, “Machine learning, a probabilistic perspective”, ed: Taylor & Francis, 2014.
[31] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, Mathematics for machine learning: Cambridge University Press, 2020.
[32] Cormen, Thomas H., et al. Introduction to algorithms. MIT press, 2022.

زیرنویس‌ها

[1] Very-Large-Scale Integration
[2] Dynamic Voltage Frequency Scaling
[3] Dynamic Power Management
[4] Decision Tree
[5] Multi-Processor System on a Chip
[6] K-Nearest Neighbors
[7] Look-Ahead Earliest Deadline First
[8] Long Short-Term Memory
[9] Seq-to-Seq Long Short-Term Memory
[10] Million Instructions Per Second
[11] Full System
[12] Multicore Power, Area, and Timing
[13] Principal Component Analysis
[14] Baysian
[15] Accuracy
[16] True Positive
[17] True Negative
[18] False Positive
[19] False Negative
[20] Sum of squared errors

[Downloaded from jiaeee.com on 2026-11-16]
[DOI: 10.61186/jiaeee.22.1.99]