Energy Efficiency of AI-powered Components: A Comparative Study of Feature Selection Methods

Rafiullah Omar*, Henry Muccini*
*FrAmeLab, University of L'Aquila, L'Aquila, Italy,
rafiullah.omar@graduate.univaq.it, henry.muccini@univaq.it

***Abstract*—Machine learning models, at their core, are software systems that demand computational resources, making them a pertinent concern for software engineers. The energy consumed by these models during training and inference phases can have far-reaching consequences, from the environmental impact to operational costs and even the user experience. Consequently, understanding how different software components, such as feature selection methods, impact energy efficiency becomes essential for software engineers tasked with building sustainable and cost-effective AI-driven solutions.**

By addressing four key research questions (RQ1-RQ4), we aim to provide software engineers with actionable insights into making informed decisions about feature selection methods to strike the right balance between energy efficiency and model accuracy.

CCS CONCEPTS - Software and its engineering → Software libraries and repositories - Computing methodologies → Machine learning

KEYWORDS Energy consumption, run-time performance, feature selection methods, environmental impact, Green AI

I. Introduction

In an era dominated by machine learning applications, the energy efficiency of these models has emerged as a critical concern with significant implications for the field of software engineering. As machine learning tasks continue to grow in scale and complexity, so does the associated energy consumption, raising important questions about the environmental and cost impacts on software systems and applications.

This research delves into the intricate relationship between *feature selection methods* and *energy efficiency* in the realm of machine learning, offering valuable insights that have direct relevance to software engineering practices. Software engineering, as a discipline, is intimately intertwined with the development, deployment, and maintenance of software systems that span a multitude of domains and industries. The rise of machine learning applications has expanded the scope of software engineering, with AI-powered systems becoming integral components of modern software solutions. However, this fusion of AI and software engineering brings forth a critical challenge: *optimizing the energy efficiency of these systems without compromising on their functionality and performance*. This study focuses on investigating how different feature selection and scoring methods impact the energy efficiency of machine learning models, a concern that directly relates to software engineering practices. Feature selection, an integral step in model development, influences not only predictive accuracy but also the computational resources required. Therefore, the findings of this research hold significant implications for software engineers who strive to create efficient and environmentally responsible software systems. Our comprehensive experimental setup considers various modification methods, scoring methods, feature quantities, and machine learning models—components that software engineers encounter in the development pipeline.

Four research questions are addressed: RQ1 examines the relative energy efficiency of two distinct univariate feature selection methods , RQ2 explores the comparative energy efficiency of various feature selection methods, RQ3 analyzes the impact of different feature selection methods on training energy consumption, and RQ4 assesses the effect of these methods on the accuracy of machine learning models. To answer these questions, a comprehensive experimental setup involving feature selection methods, the number of features, and machine learning models is employed. The analysis reveals that while feature modification methods for univariate features exhibit no significant differences, the choice of feature selection method and scoring methods for univariate features selection significantly impacts energy consumption and model accuracy. RFE and sequestial feature selection methods are energy-intensive but may yield better model performance in limited specific cases.

These findings not only advance the understanding of energy-efficient machine learning techniques but also provide a direct bridge to software engineering practices, guiding developers, architects, and decision-makers in the pursuit of sustainable and cost-effective software solutions. However, it is essential to recognize that hardware constraints, dataset characteristics, and real-world conditions may affect the applicability of these findings in software engineering contexts. Nevertheless, this research lays a foundation for the ongoing integration of energy-efficient AI into software engineering, aligning with the broader goals of sustainability and responsible technology development.

II. Experimental Setup

A. Goals

The primary goal of this research is to investigate and enhance the energy efficiency of feature selection methods in machine learning. By addressing key research questions and conducting experiments, our aim is to provide practitioners with valuable insights and guidelines for achieving a balance between energy efficiency and model accuracy in machine learning applications. This research also serves as a foundation for future studies aimed at advancing the field of sustainable and cost-effective AI.

B. Research Questions

- **RQ1. What is the relative energy efficiency of two distinct univariate feature selection methods in machine learning?** Different feature selection methods are available for classification tasks. We selected the methods implemented in the popular ML library scikit-learn [1]. To ensure a fair comparison, we focused on methods within a single standard library, as comparing methods from different libraries can introduce variations in energy consumption due to implementation differences [31]. For univariate feature selection, scikit-learn offers two primary methods: SelectKBest: This method selects a specified number of top features based on a scoring function. SelectPercentile: This method selects a specified percentage of top features based on a scoring function. Both methods accept scoring methods as parameters. The key difference lies in how they define the number of features to retain: SelectKBest uses a fixed number, while SelectPercentile uses a percentage. We'll explore three feature scoring methods: Chi-squared f_classif mutual_info_classif Each scoring method will be used with both SelectKBest and SelectPercentile. RQ1 will investigate the energy efficiency of these combinations. Based on the results, we'll select the most energy-efficient method and implement different scoring methods within it. Furthermore, we'll consider the three scoring methods themselves as feature selection methods for comparison with other methods in our study and to maintain consistency with terminology used in related research [32].
- **RQ2. What is the comparative energy efficiency of various feature selection methods in machine learning?** This research question aims to compare the energy efficiency of various feature selection methods in machine learning. Specifically, three feature scoring methods within univariate feature selection are considered as part of these methods. The objective is to understand the impact of these methods on energy consumption, providing insights for optimizing energy resources in machine learning workflows.
- **RQ3. How do different feature selection methods impact the training energy consumption of machine learning models?** This research question explores the influence of various feature selection methods on the training energy consumption of machine learning models. By examining these impacts, it sheds light on how different methods affect the energy efficiency of model training processes.
- **RQ4. How do different feature selection methods impact the accuracy of machine learning models?** This research question delves into the influence of different feature selection methods on the accuracy of machine learning models. It seeks to uncover how these methods affect model performance, providing valuable insights for optimizing the trade-off between accuracy and energy efficiency in machine learning.

C. Variables

There exist four independent variables (IVs) and three dependent variables (DVs) within our research framework. These independent variables are delineated as follows:

1) Modification Methods (IV1): Within the standard scikit-learn 1.1.2 Python library, two distinct modification methods are implemented for univariate feature selection. The first method, known as 'SelectPercentile' selects a specified percentage of features from the total set, while the second method, 'SelectKBest' accepts a predefined number of features to select from the dataset.
2) Feature selection Methods (IV2): For classification tasks, various feature selection methods are implemented in scikit-learn. We specifically chose methods from the scikit-learn library to ensure a consistent comparison. Selecting methods from different libraries could introduce difficulties in comparison due to the potential impact of the library on energy consumption [31]. The chosen methods are:
   a) **Removing features with low variance:** It removes all features whose variance doesn't meet some threshold, We calculate the threshold based on the feature percentage.
   b) **Select from model:** This method uses an ML model to assign importance to each feature and subsequently removes features with low importance. We employed the Random Forest Classifier for this purpose due to its direct access to feature importance information after training.
   c) **Mutual information:** Mutual information (MI) between two random variables is a non-negative value that measures the dependency between the variables [2].
   d) **Chi-square:** It computes chi-squared statistics between each non-negative feature and class[2].
   e) **if_classif:** This method calculates the ANOVA F-value for the provided sample[2].
3) Number of Features (IV3): To assess the energy required for dataset modification and the subsequent energy consumption during model training, we have selected various percentages of features, ranging from 10% to 100%.
4) Machine Learning Models (IV4): Six machine learning models were chosen for our experiments, encompassing SVM, KNN, AdaBoost, Decision Tree, Random Forest,

[1]https://scikit-learn.org/stable/modules/feature_selection.html

[2]https://scikit-learn.org/stable/modules/feature_selection.html

and Bagging Classifier. our focus is on the energy consumption of the feature selection methods and their impact on the ML models accuracy and energy consumption of ML models during training, that is why we don't take into consideration the hyperparameters tuning and used the standard hyperparameters as defined in the library.

The dependent variables are defined as follows:

1) Amount of Energy Consumed for Modifying the Dataset (DV1): During the modification of the dataset using different modification and scoring methods, we meticulously record the quantity of energy expended for these modifications.
2) Amount of Energy Consumed for Training the Model (DV2): After dataset modification, we proceed to train the model, meticulously recording the energy consumption throughout this process for the modified dataset.
3) Accuracy (DV3): In order to gauge model performance, we record the accuracy of the trained models upon the modified dataset. This assessment of accuracy is quantified through the F1 score metric.

D. Experimental setting

To investigate these four research questions, we selected a publicly available dataset that has been previously employed in similar studies [32]. This dataset was sourced from the University of California, Irvine's machine learning repository [3] preprocessing utilizing the term frequency–inverse document frequency (tf-idf) technique, and we implemented standard message tokenization using the functionality provided by scikit-learn version 1.1.2. Post-tokenization, the dataset expanded to encompass a total of 8,168 features. The experiments were conducted on a core i7 system with 16GB of RAM, and energy measurement was carried out using codecarbon[4] version 2.1.4, a Python open-source library designed for quantifying the energy consumption of machine learning models. The Python version employed throughout this research is 3.10.11.

E. Experimental procedure

To address the first research question comprehensively, we designed an extensive experimental setup, encompassing a total of 1,200 individual experiments. These experiments were meticulously executed, with each one being repeated 20 times to ensure the reliability and consistency of our results. This rigorous repetition process enabled us to amass a substantial dataset that would provide a robust foundation for our analysis. Throughout these experiments, we recorded all necessary parameters for the analysis as listed: feature modification method, algorithm, experiment id, iteration number, number of features, preprocessing energy in joules, preprocessing time in seconds, train energy in joules, train time in seconds, predict energy in joules, predict time (seconds), data type, accuracy, precision, recall, F1 score, experiment date and time. For the subsequent research questions, pertaining to questions 2 and 3, we embarked on an even more ambitious endeavor. We conducted a multitude of experiments, exploring various permutations and combinations of scoring methods and machine learning algorithms. With six distinct feature selection methods, 10 percentages, and six machine learning algorithms at our disposal, the sheer number of potential combinations to explore was vast. As a result, our research involved a staggering 7,200 experimental runs. This extensive experimentation allowed us to comprehensively examine the impact of different scoring methods and modification techniques on both energy consumption and model performance. Moreover, we maintained our commitment to accuracy and reliability by repeating each of these experiments 20 times. This meticulous and exhaustive approach to experimentation not only ensured the robustness of our findings but also provided a wealth of data for in-depth analysis. By investing substantial effort into the experimental phase, we aimed to offer valuable insights into the intricate relationships between feature selection methods, energy efficiency, and machine learning model performance.

F. Analysis

In this section we report the adopted procedure for the data analysis. we employed rigorous statistical methods. Specifically, we applied the Mann-Whitney U test to our experimental data. This choice of statistical test was guided by a preliminary examination of the data's normality using Shapiro-Wilk normality test, as we check the effects of different feture selection methods on different ML algorithms's training energy consumption so we run the test on all groups of data. The results of the test, with a test statistic of laess than 0.89 and p value 1.52e-12 for all groups, unequivocally indicated that the data from these experiments did not conform to a normal distribution. This led us to the prudent decision of employing non-parametric testing, such as the Mann-Whitney U test, which is robust in the presence of non-normal data. To understand the effect size we calculated the Cohen's d value.

III. Result

A. Answer to RQ1

In order to determine the most energy-efficient method, our analysis of the data generated from multiple experiments, as depicted in Figure 1, reveals that there is no significant difference in the energy consumption between the two feature selection methods. The observations in Figure 1 are further substantiated by the Mann–Whitney U test result, p-value is 0.8, which exceeds the significance level of 0.05, as determined through the Mann-Whitney U test applied to the experimental data. Consequently, due to the SelectPercentile method's capability to select features based on a specified percentage, and its ease of use, we have opted to proceed with this feature selection method in a subsequent set of experiments for Univariate feature selection in RQ2 and RQ3.

[3] https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection. Accessed 21th July 2023

[4] https://codecarbon.io/

Answer to RQ1: Though energy consumption showed no significant difference between SelectKBest and SelectPercentile, the ability of SelectPercentile to select features by percentage led to its choice for further experiments.

B. Answer to RQ2

In addressing this question, our objective is to identify the most energy-efficient feature selection method. As visually depicted in Figure 2, it becomes evident that the Recursive Feature Elimination (RFE) method exhibits the highest energy consumption, while 'f_classif' emerges as the most energy-efficient among the six feature selection methods which save 99.99% energy compared to RFE. After RFE, the second method that consumes more energy is the Mutual Information scoring method, used with the SelectPercentile feature selection method, as shown in Table I. The mean difference between RFE and the Mutual Information method is 83.31%. Notably, RFE differs from other methods, as depicted in the figure, where an increase in the percentage of features to select results in a reduction in energy consumption.

After Mutual Information, the third method that consumes more energy is SelectFromModel, which consumes 83.17% less energy than Mutual Information. As it utilizes a model for assigning importance to the features, its energy consumption is dependent on the model used for assigning importance. As discussed in the experiments design section, we selected the RandomForestClassifier [32] model due to its lower energy consumption compared to other models with direct access to the features' importance property after training [32].

The fourth method that consumes less energy is Variance Threshold, consuming 78.22% less energy compared to SelectFromModel. The fifth method that consumes less energy compared to the aforementioned methods is the Chi-square (chi2) method, consuming 97.67% less energy compared to Variance Threshold. Lastly, the most energy-efficient method is 'f_classif,' consuming 31.79% less energy compared to 'chi2'. In summary, based on the Mann-Whitney U test results and the percentage of mean difference in energy consumption among different methods, there is a significant difference between the energy consumption of various feature selection methods. The minimum difference is between 'f_classif' and 'chi2,' indicating that using 'f_classif' can save 31.79% energy compared to 'chi2', while the most significant difference is between 'f_classif' and RFE, suggesting that choosing 'f_classif' over RFE can save 99.99% energy. Additionally, Cohen's d value between these methods further supports this large effect size, indicating a substantial difference in their energy consumption.

Answer to RQ2: RQ2 reveals significant discrepancies in energy consumption among different methods. "f_classif" emerges as the clear winner, consuming an impressive 99.99% less energy than the least efficient option, RFE.

C. Answer to RQ3

In RQ3, our objective is to gain insight into the impact of scoring methods on energy consumption during the model training phase. It is well-established that reducing the number of features can lead to energy savings during training, as evidenced by the findings illustrated in Figure 3. This visual representation demonstrates that machine learning models with a reduced feature set tend to exhibit lower energy consumption across most model types, with the exception of KNN. We can also see in Figure 3 that different scoring methods have different effects on the energy consumption of some models during model training, such as in bagging classifier, and decision tree. In bagging classifier and decision tree, 'f_classif' performs well compared to other methods, it save 45% energy compared to variance threshold, and also in the decision tree, it consumed 31% less energy compared to varinace threshold. Variance threshold in both nodels consume more energy. While in SVM, the RFE is an energy-efficient method during the training phase. In Random Forest, we see that MI is the energy-efficient method, but in Adabost, the difference is not much obvious from the graph. These observations align with the results of the Mann-Whitney U test statistics and p-values computed for each model. The test results are available in the replication package[5].

Answer to RQ3: In Bagging, Decision Tree, and Random Forest f_classif and MI are energy efficent methods.

D. Answer to RQ4

Figure 4 visually illustrates no significant discrepancies in model accuracy across datasets modified using different feature selection methods. However, the Mann-Whitney U test revealed statistically significant differences in the accuracy of KNN trained on datasets modified by different feature selection techniques. Crucially, feature reduction did not yield a reduction in energy consumption during KNN training. Moreover, training KNN on smaller datasets negatively impacted its accuracy without offering energy savings. Therefore, we conclude that feature reduction is not recommended for the KNN model in the context of energy conservation and model accuracy.

Answer to RQ4: No significant difference except in KNN.

[5]https://doi.org/10.5281/zenodo.10612801

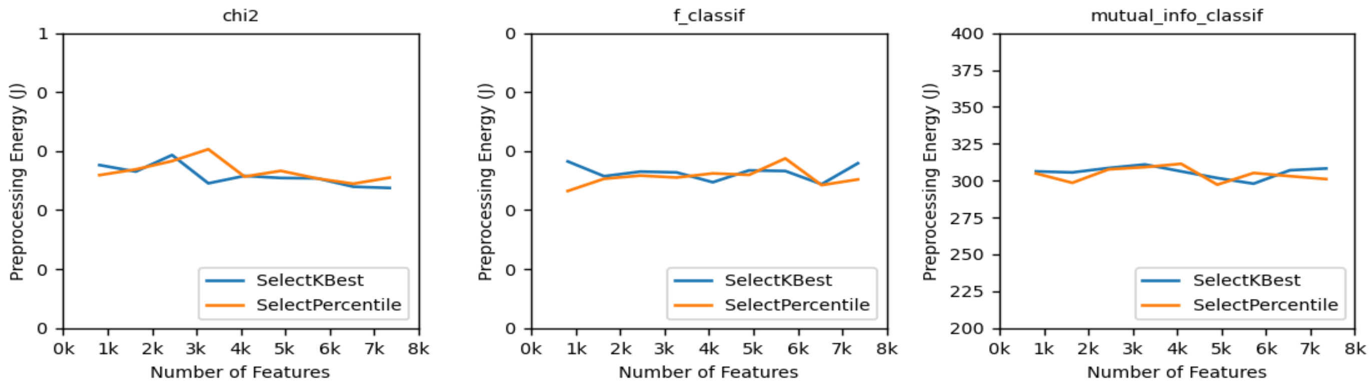

Fig. 1. Energy consumption of SelectKBest and SelectPercentile.

TABLE I
MODIFICATION METHODS ENERGY CONSUMPTION ALPHA=0.05

M1	M2	p_value	M1 mean	M2 mean	Difference (%)	Cohen_Value
RFE	f_classif	2.31E-119	1957.1	0.3	99.99	4.0
RFE	chi2	3.46E-119	1957.1	0.4	99.98	4.0
RFE	VarianceThreshold	4.05E-78	1957.1	12.0	99.39	2.7
RFE	SelectFromModel	5.28E-79	1957.1	55.0	97.19	2.6
RFE	mutual_info_classif	6.73E-120	1957.1	326.6	83.31	3.2
mutual_info_classif	f_classif	2.30E-257	326.6	0.3	99.91	3.5
mutual_info_classif	chi2	7.15E-257	326.6	0.4	99.88	3.5
mutual_info_classif	VarianceThreshold	1.75E-119	326.6	12.0	96.33	2.7
mutual_info_classif	SelectFromModel	6.39E-121	326.6	55.0	83.17	2.3
SelectFromModel	f_classif	1.80E-119	55.0	0.3	99.49	6.1
SelectFromModel	chi2	2.34E-119	55.0	0.4	99.26	6.1
SelectFromModel	VarianceThreshold	2.95E-78	55.0	12.0	78.22	3.2
VarianceThreshold	f_classif	5.92E-123	12.0	0.3	97.67	5.2
VarianceThreshold	chi2	5.57E-121	12.0	0.4	96.59	5.1
chi2	f_classif	4.31E-135	0.4	0.3	31.79	1.1

Test results indicate that the most significant difference in accuracy obtained with different feature selection methods across various machine learning models is 0.014, observed in the decision tree between RFE and Variance Threshold. We achieve better accuracy with RFE compared to Variance Threshold in the decision tree model. Consequently, the selection of methods for the decision tree will be solely based on accuracy requirements and energy trade-offs. KNN is the only model significantly affected in its accuracy when reducing the number of features with any of the methods. Full details about the test results are available in the replication package.

IV. DISCUSSION

The results of our experimental study provide evidence of a substantial difference in the energy consumption among various feature selection methods. We discuss these results and offer conclusions and guidelines for practitioners and developers regarding the selection of these methods during the development of ML-enabled software.

A. *Energy Efficiency of Different Feature Selection Methods:*

Previous work closely related to ours [32] concluded that reducing the number of features during the training of machine learning models has a significant impact. Modifying features itself is a process that introduces a new source of energy consumption. Therefore, we must ensure that the energy saved

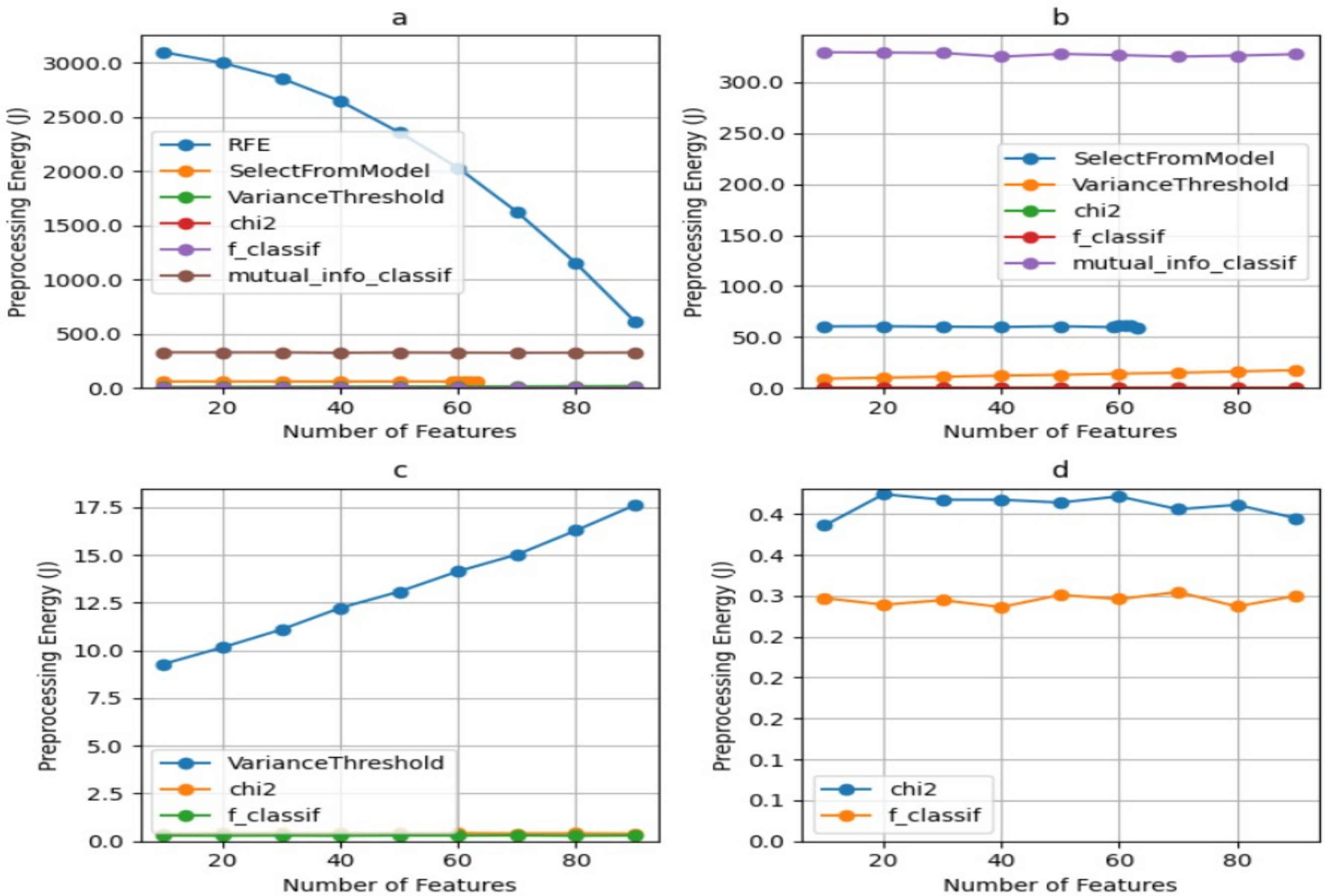

Fig. 2. Energy consumption of feature selection methods.

during the training phase of machine learning models exceeds the energy consumed for feature selection. In some cases, if we repeatedly retrain the models, the effects of feature reduction become more pronounced. Retraining a model may occur due to the availability of new data or concept drift, necessitating a rerun of the feature selection process. It is not guaranteed that previously selected features will retain the same level of importance. Thus, by reducing the energy required for the feature selection process, we effectively increase overall energy savings. Based on our results, Recursive Feature Elimination (RFE) is one of the methods that consumes more energy. When comparing its energy consumption during feature selection with the energy consumed by the decision tree during training, RFE consumes 4000 times more energy than the training phase. If this difference is substantial, especially in scenarios where new data is received frequently, and the model is retrained upon data arrival, we may inadvertently introduce an additional source of energy consumption that surpasses the energy saved. Careful consideration and optimization of the feature selection process are crucial to ensuring a net gain in energy efficiency. Several factors influence the energy consumption of 'RFE'. Firstly, the estimator it uses for feature removal requires retraining each time a feature is removed. Despite selecting an energy-efficient machine learning model, the maximum iteration it undergoes contributes to increased energy consumption. Another factor affecting the energy consumption is the number of features removed with each iteration. If this number is small, numerous iterations occur, leading to higher energy consumption. Conversely, removing a large number of features with each iteration may impact the model's accuracy, as important features may be discarded. Furthermore, the total number of features in a dataset plays a crucial role, as 'RFE' utilizes a model, and the model's energy consumption is potentially dependent on the number of features [32]. Thus, we can conclude that using 'RFE' with a dataset containing a large number of features, where models are frequently trained, may not be a favorable decision in terms of energy consumption. f_classif is the most energy efficient method among the compared methods.

B. *Effects of These Methods on Model Training Energy Consumption*

To further investigate whether the choice of feature selection methods influences the energy consumption of machine learning (ML) models during training. When different models are trained on datasets modified with various methods, we discovered significant differences among some methods for specific models. In the bagging classifier, the mutual infor-

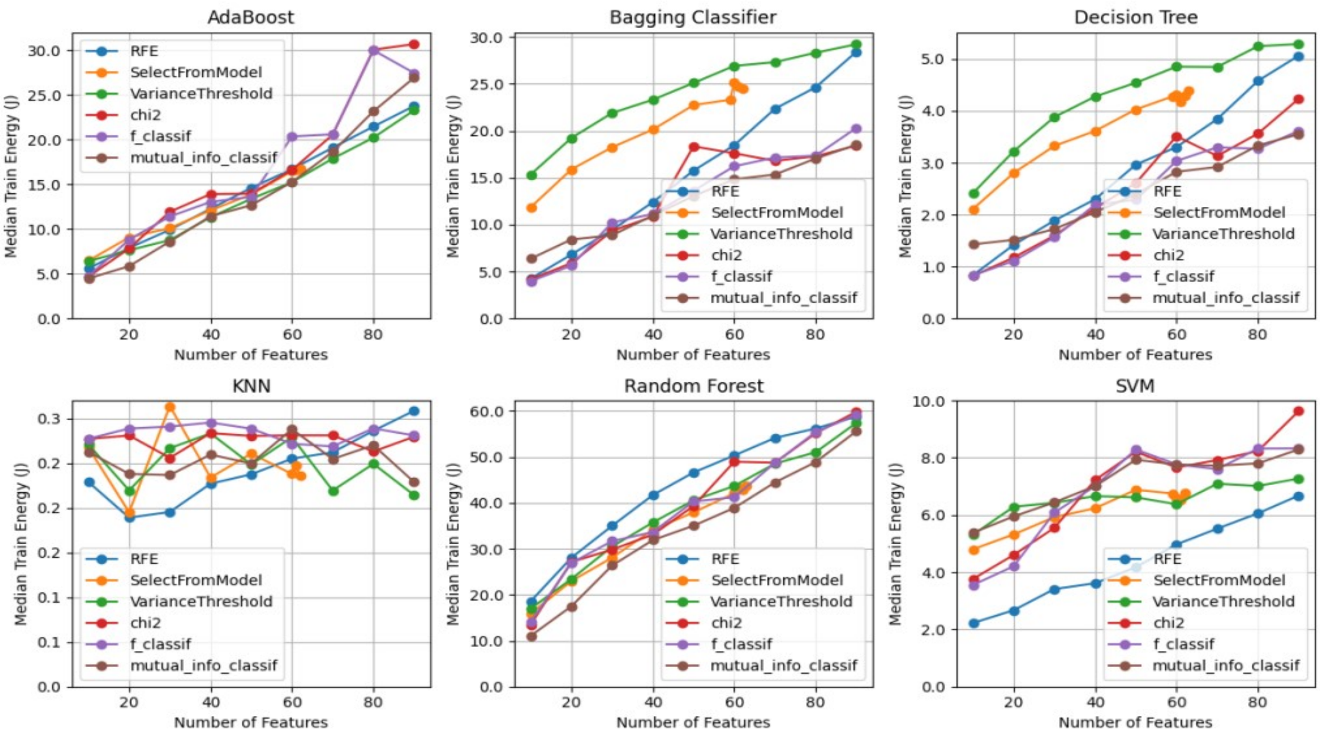

Fig. 3. Models training Energy consumption.

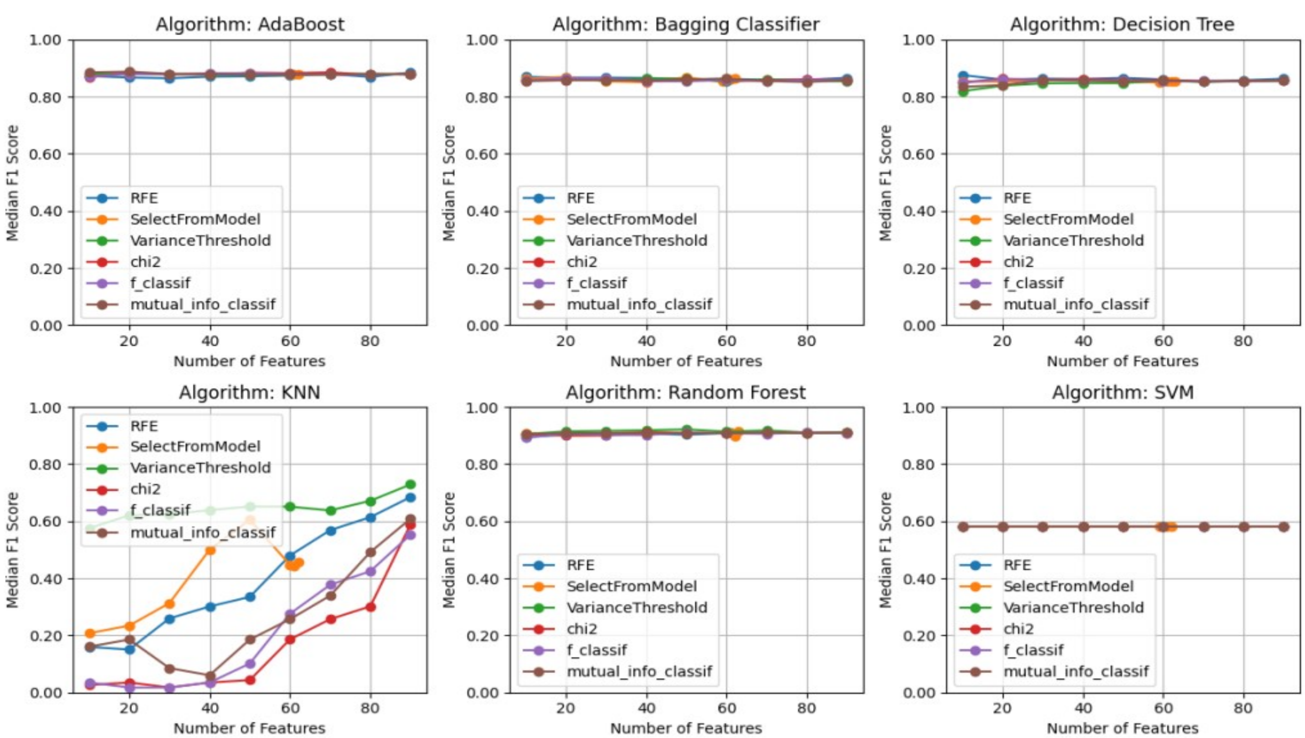

Fig. 4. Models accuracy.

mation method saves 49.9% more energy compared to the variance threshold. In SVM, RFE saves 43% more energy than chi2. For decision trees, mutual information saves 43% more energy than the variance threshold. In Adaboost, select from model saves 29.8% more energy compared to chi2. Detailed information about these differences is available in the replication package. This information can assist developers in making decisions based on the frequency of retraining, availability of new data, number of reruns of feature selection methods, and the energy consumption of these methods. Subsequently, developers can opt for an energy-efficient feature selection method.

C. Effects of These Methods on the Models' Accuracy

KNN is one of the models for which reducing features does not affect its energy during training, but it does impact its accuracy. In contrast, there is no significant difference between the feature selection methods in other models. The maximum difference observed in the results is 0.01, which is between chi2 and variance threshold in the decision tree model. Additionally, there is a 0.009 difference between f_classif and variance threshold, where, in both cases, variance threshold achieves lower accuracy compared to the other two methods. This outcome is favorable because variance threshold consumes more energy during the feature reduction process. Full details are available in the replication package, the remaining differences in the accuracy are less than 0.009.

V. Threats to Validity

A potential challenge to internal validity, linked to historical factors, may have arisen in our experiment due to the potential impact of executing successive iterations on our measurements, e.g., due to rising hardware temperatures. To address this concern, we implemented measures by introducing a 5-second sleep operation before each experimental iteration. This ensured more uniform hardware conditions for all runs. Likewise, a warm-up operation was conducted to guarantee that the initial iteration occurred under very similar conditions to subsequent ones, mitigating potential influences on our measurements. As a threat to reliability of measure, the presence of background tasks during the experiment execution could have served as confounding factors, thereby affecting our energy measurements. To address this concern, we took preemptive measures by terminating processes that were not essential for the execution of the experiment and restricted access to the infrastructure. Furthermore, we conducted each experiment 20 times to minimize the impact of any unforeseen background processes. To ensure the reproducibility of our study, we have made the replication package accessible online [6]. Running the experiments on different hardware yielded consistent results, reinforcing the reliability of our findings and offering assurance in the robustness of the outcomes.

VI. Related Work

In prior research, energy consumption in the context of software systems has garnered significant attention, with a focus on various domains [20], [21], levels, and ecosystems [11]. Notably, some have explored the energy consumption of different programming languages and different data structures [17]–[19] and developed some tools for the making the applications green (er) [22], [23] ,some studies have explored the energy efficiency of AI-based systems, albeit within a limited scope. For instance, prior work has delved into AI's substantial energy requirements and its environmental and financial implications [2], [3], [8], [10], [13], [25]. Researchers have examined practices to enhance traditional machine learning (ML) methods, aiming to reduce energy consumption [6], [7], [9], [12]. These investigations have yielded insights into factors affecting energy use in ML for specific applications, such as Android devices. Additionally, studies have proposed guidelines and models for estimating energy consumption in ML applications [4], [5], [26]. Another line of research has evaluated the energy efficiency of AI models, aiming to make them more sustainable. Techniques such as reducing unnecessary computations in convolutional neural networks (CNNs) have demonstrated significant energy savings [13], [14]. Researchers have also developed models and tools to estimate energy consumption in ML applications [4], [5], [29]. They have highlighted the challenges of relying solely on ML models for such estimations [1]. In parallel, some studies have begun addressing energy consumption in AI and machine learning with a broader perspective. These studies have drawn attention to the substantial environmental impact of training large AI models and introduced the concept of Green AI, which considers energy consumption as a critical performance metric alongside accuracy [15]. They emphasize the importance of researching strategies to reduce the energy footprint of AI systems. Furthermore, research on AI sustainability has gained momentum, with a focus on energy-efficient AI development [16]. These studies offer recommendations for mitigating the growth of energy-intensive AI models [16], including the curation of datasets and considerations of potential risks in AI development. They also underscore the need for increased research on Green AI and its potential impact on the sustainability of AI projects [15]. Another study investigated the energy efficiency of different DL models implemented in various AI frameworks [31], while study [32] explores the utilization of data-centric approaches to reduce the amount of energy needed for training ML models. In [32], a data-centric approach is studied with a specific focus on reducing the size of the dataset and its impact on the energy consumption of ML models during training, while considering model accuracy. Their results indicate potential energy savings of up to 92.16% by reducing the dataset size. Notably, they reduced the dataset in terms of features, employing the chi-square method for features reduction. While alternative methods exist for feature reduction, it is essential to make informed decisions considering the energy cost associated with the size reduction process. To the best of our knowledge, no existing literature investigates the energy efficiency of available methods for dataset size reduction. Addressing this gap, our study bridges these research domains by empirically comparing the energy consumption of various feature selection methods in machine learning. We delve into the energy efficiency of different feature selection methods, their impact on ML model accuracy,

[6]https://doi.org/10.5281/zenodo.10612801

and their implications for energy consumption during both training and inference phases.

This work extends the current research landscape by shedding light on data-centric approaches aimed at achieving Green AI and reducing the energy demands of machine learning applications [30].

VII. Future Work

This research has provided valuable insights into the energy efficiency of feature selection methods in machine learning. However, there are several avenues for future work that can further advance our understanding and contribute to the development of more energy-efficient machine learning practices:

1) Dynamic Feature Selection: Investigating the feasibility of dynamically adapting feature selection methods during the machine learning process could be beneficial. By dynamically selecting features based on model performance and energy consumption, practitioners could achieve a better balance between accuracy and efficiency.
2) Benchmarking on Diverse Datasets: Expanding the study to include diverse datasets from various domains can provide a more comprehensive understanding of the relationship between feature selection methods, energy efficiency, and model performance. Different datasets may exhibit distinct behaviors regarding energy consumption.
3) Real-world Applications: Applying the findings from this research to real-world applications and case studies in different industries, such as healthcare, finance, or environmental monitoring, can validate the practical implications of energy-efficient feature selection methods.

VIII. Conclusion

Our experimental study has revealed substantial differences in the energy consumption of various feature selection methods, providing crucial insights for practitioners and developers involved in machine learning (ML)-enabled software development. The discussion below summarizes our key observations and provides actionable insights. Our investigation into the energy efficiency of different feature selection methods highlighted significant considerations. Previous work, closely aligned with our study [32], emphasized the substantial impact of reducing the number of features during ML model training. Recursive Feature Elimination (RFE), while widely used, emerged as one of the methods with higher energy consumption. On the contrary, f_classif demonstrated superior energy efficiency among the compared methods. Examining the impact of feature selection methods on ML model training energy consumption revealed distinctive patterns. Significant differences were observed across methods for specific models. Notably, mutual information in the bagging classifier, RFE in SVM, mutual information in decision trees, and select from model in Adaboost exhibited energy-saving advantages over alternative methods. These nuances provide developers with valuable insights for decision-making based on factors such as retraining frequency, data availability, and the energy consumption of feature selection methods. The accuracy of ML models was scrutinized concerning the choice of feature selection methods. While KNN demonstrated resilience to feature reduction in terms of energy, its accuracy was affected. In contrast, other models showed no significant differences among feature selection methods, except for marginal discrepancies. The replication package contains detailed information, aiding developers in making informed decisions. In conclusion, our study contributes to the understanding of energy-efficient practices in ML development. Developers are encouraged to consider the specific characteristics of their datasets and models when selecting feature selection methods, keeping a delicate balance between energy efficiency and model accuracy.

References

[1] J. Bornholt, T. Mytkowicz, and K. S. McKinley. 2012. The model is not enough: Understanding energy consumption in mobile devices. Hot Chips, 1–3

[2] A. Canziani, Adam Paszke, and E. Culurciello. 2016. An Analysis of Deep Neural Network Models for Practical Applications. ArXiv abs/1605.07678 (2016).

[3] Zhenpeng Chen, Yanbin Cao, Yuanqiang Liu, Haoyu Wang, Tao Xie, and Xuanzhe Liu. 2020. A comprehensive study on challenges in deploying deep learning based software. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020). Association for Computing Machinery, New York, NY, USA, 750–762. https:/doi.org/10.1145/3368089.3409759

[4] Cuijiao Fu, Depei Qian, and Zhongzhi Luan. 2018. Estimating Software Energy Consumption with Machine Learning Approach by Software Performance Feature. In 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data. (SmartData). 490–496. https:/doi.org10.1109Cybermatics_2018.2018.00106

[5] Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. 2019. Estimation of energy consumption in machine learning. J. Parallel and Distrib. Comput. 134 (2019), 75–88. https://doi.org/10.1016/j.jpdc.2019.07.007.

[6] Edward Y. Y. Kan, W. K. Chan, and T. H. Tse. 2012. EClass: An execution classification approach to improving the energy-efficiency of software via machine learning. Journal of Systems and Software 85, 4 (2012), 960–973. https: //doi.org/10.1016/j.jss.2011.11.1010 [7] Mohit Kumar, Xingzhou Zhang, Liangkai Liu, Yifan Wang, and Weisong Shi. 2020. Energy-Efficient Machine Learning on the Edges. In 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). 912–921. https://doi.org/10.1109/IPDPSW50202.2020.00153

[7] Da Li, Xinbo Chen, Michela Becchi, and Ziliang Zong. 2016. Evaluating the Energy Efficiency of Deep Convolutional Neural Networks on CPUs and GPUs. In 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom). 477–484. https://doi.org/10.1109/BDCloud-SocialCom-SustainCom.2016.76

[8] Andrea McIntosh, Safwat Hassan, and Abram Hindle. 2018. What can Android mobile app developers do about the energy consumption of machine learning? Empirical Software Engineering (10 May 2018), 1–42.

[9] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. Commun. ACM 63, 12 (Nov. 2020), 54–63. https://doi.org/10.1145/3381831

[10] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. Commun. ACM 63, 12 (Nov. 2020), 54–63. https://doi.org/10.1145/3381831

[11] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 3645–3650. https://doi.org/10.18653/ v1/P19-1355

[12] Yue Wang, Ziyu Jiang, Xiaohan Chen, Pengfei Xu, Yang Zhao, Yingyan Lin, and Zhangyang Wang. 2019. E2-Train: Training State-of-the-art CNNs with Over 80% Energy Savings. Advances in Neural Information Processing Systems 32 (2019). https://proceedings.neurips.cc/paper/2019/hash/663772ea088360f95bac3dc7ffb841be-Abstract.html

[13] Xingzhou Zhang, Yifan Wang, and Weisong Shi. 2018. pCAMP: Performance Comparison of Machine Learning Packages on the Edges. In USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). USENIX Association, Boston, MA. https://www.usenix.org/conference/hotedge18/presentation/zhang

[14] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," 6 2019. [Online]. Available: http://arxiv.org/abs/1906.02243

[15] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" Association for Computing Machinery, Inc, 3 2021, pp. 610–623.

[16] C. Calero, M. Polo, and M. A. Moraga, "Investigating the impact on ´ execution time and energy consumption of developing with spring," Sustainable Computing: Informatics and Systems, vol. 32, p. 100603, 2021.

[17] W. Oliveira, R. Oliveira, F. Castor, B. Fernandes, and G. Pinto, "Recommending energy-efficient java collections," in 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). IEEE, 2019, pp. 160–170.

[18] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. P. Fernandes, and J. Saraiva, "Ranking programming languages by energy efficiency," Science of Computer Programming, vol. 205, p. 102609, 2021.

[19] S. Georgiou, M. Kechagia, P. Louridas, and D. Spinellis, "What are your programming language's energy-delay implications?" in Proceedings of the 15th International Conference on Mining Software Repositories, 2018, pp. 303–313.

[20] H. Anwar, B. Demirer, D. Pfahl, and S. Srirama, "Should energy consumption influence the choice of android third-party http libraries?" in Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems, 2020, pp. 87–97.

[21] R. Verdecchia, G. Procaccianti, I. Malavolta, P. Lago, and J. Koedijk, "Estimating energy impact of software releases and deployment strategies: The kpmg case study," in 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). IEEE, 2017, pp. 257–266.

[22] A. Ribeiro, J. F. Ferreira, and A. Mendes, "Ecoandroid: An android studio plugin for developing energy-efficient java mobile applications," 2021.

[23] L. Cruz and R. Abreu, "Emaas: Energy measurements as a service for mobile applications," in 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), 2019, pp. 101–104.

[24] A. van Wynsberghe, "Sustainable AI: AI for sustainability and the sustainability of AI," AI and Ethics, pp. 1–6, 2021. [20] V. Kloh, B. Schulze, and M. Ferro, "Use of machine learning for ˆ improvements in performance and energy consumption in hpc systems," 09 2020.

[25] S. Zhu, K. Ota, and M. Dong, "Green AI for IIoT: Energy efficient intelligent edge computing for industrial internet of things," IEEE Transactions on Green Communications and Networking, 08 2021.

[26] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. A. Behram, J. Huang, C. Bai, M. Gschwind, A. Gupta, M. Ott, A. Melnikov, S. Candido, D. Brooks, G. Chauhan, B. Lee, H.-H. S. Lee, B. Akyildiz, M. Balandat, J. Spisak, R. Jain, M. Rabbat, and K. Hazelwood, "Sustainable AI: Environmental implications, challenges and opportunities," 10 2021. [Online]. Available: http://arxiv.org/abs/2111.00364

[27] G. Yang, E. Hu, I. Babuschkin, S. Sidor, X. Liu, D. Farhi, N. Ryder, J. Pachocki, W. Chen, and J. Gao, "Tuning large neural networks via zero-shot hyperparameter transfer," Advances in Neural Information Processing Systems, vol. 34, 2021.

[28] F. Zogaj, J. P. Cambronero, M. C. Rinard, and J. Cito, "Doing more with less: characterizing dataset downsampling for automl," Proceedings of the VLDB Endowment, vol. 14, no. 11, pp. 2059–2072, 2021.

[29] E. Garcia-Martin, N. Lavesson, and H. Grahn, "Identification of energy hotspots: A case study of the very fast decision tree," Green, Pervasive, and Cloud Computing, pp. 267–281, 01 2017.

[30] A. McIntosh, S. Hassan, and A. Hindle, "What can android mobile app developers do about the energy consumption of machine learning?" Empirical Software Engineering, pp. 1–42, May 2018.

[31] S. Georgiou, M. Kechagia, T. Sharma, F. Sarro, and Y. Zou, "Green AI: Do Deep Learning Frameworks Have Different Costs?" in 2022 IEEE/ACM 44st International Conference on Software Engineering (ICSE), 2022.

[32] R. Verdecchia, L. Cruz, J. Sallou, M. Lin, J. Wickenden, and E. Hotellier, "Data-Centric Green AI: An Exploratory Empirical Study," in 2022 International Conference on ICT for Sustainability (ICT4S), Los Alamitos, CA, USA, Jun. 2022, pp. 35-45. DOI: 10.1109/ICT4S55073.2022.00015.