Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Energy-Efficient Cascaded Machine Learning Framework for Predictive Network Anomaly Detection

Dileesh Chandra Bikkasani

dbikkasa@my.bridgeport.edu

University of Bridgeport

Research Article

Keywords: Network Anomaly Detection, Machine Learning, Energy-Efficient Computing, Intrusion Detection System, Green AI, Sustainable Network Security, Cascaded Classifier

Posted Date: April 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9164110/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

The escalating complexity and scale of modern network infrastructures have amplified the need for intelligent intrusion detection systems (IDS) capable of identifying malicious traffic in real time. The growing environmental footprint of computational processes demands that such systems operate within sustainable energy budgets. This paper presents GreenShield, a two-stage cascaded machine learning framework designed to achieve high detection accuracy while minimizing computational energy consumption. The framework employs a lightweight Decision Tree classifier as the first-stage filter to handle high-confidence predictions efficiently, escalating only uncertain samples to a more resource-intensive Random Forest classifier in the second stage. Comprehensive experiments are conducted on a network traffic dataset modeled after the NSL-KDD benchmark, evaluating six machine learning classifiers: Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine, Logistic Regression, and K-Nearest Neighbors. Each model is assessed across multiple dimensions, including detection accuracy, precision, recall, F1-score, training time, inference latency, memory consumption, and estimated energy usage measured in milliwatt-hours. The proposed GreenShield framework achieves an F1-score of 0.9687, comparable to the standalone Random Forest (0.9704), while requiring significantly less computational energy for inference by resolving 98.5% of samples at the lightweight first stage. A sustainability efficiency metric is introduced to quantify the trade-off between classification performance and energy cost. Results demonstrate that GreenShield offers a favorable balance between accuracy and sustainability, making it suitable for deployment in energy-constrained network environments. This work contributes to the growing body of research on Green AI by providing evidence that cascaded classification architectures can reduce the environmental impact of network security operations without sacrificing protective capability.

1 Introduction

The rapid proliferation of interconnected devices and the expansion of network infrastructures, encompassing Internet of Things (IoT) ecosystems, cloud computing platforms, and edge computing architectures, have fundamentally transformed the digital landscape. While these advancements offer unprecedented opportunities for innovation and connectivity, they have simultaneously introduced an increasingly complex threat surface that adversaries exploit through sophisticated cyberattacks [1, 2]. Network intrusion detection systems (IDS) serve as a critical line of defense, monitoring network traffic to identify and mitigate malicious activity. Traditional rule-based and signature-based IDS approaches, while effective against known threats, struggle to detect novel or zero-day attacks, motivating the adoption of machine learning (ML) techniques that can learn patterns from data and generalize to previously unseen attack vectors [3, 4].

Machine learning has demonstrated considerable promise in network anomaly detection, with algorithms such as Decision Trees [5], Random Forests [6], Support Vector Machines [7], and ensemble methods [8] achieving high classification accuracy on benchmark datasets. However, the deployment of these models in production environments raises an important but often overlooked concern: the

computational and energy costs associated with training and inference. As highlighted by Strubell et al. [9] and Schwartz et al. [10], the carbon footprint of machine learning operations can be substantial, and the concept of Green AI has emerged as a call to develop ML systems that prioritize efficiency alongside accuracy.

The intersection of network security and sustainability presents a compelling research challenge. Network monitoring systems operate continuously, processing vast volumes of traffic data around the clock. Even modest improvements in the energy efficiency of these systems can yield significant cumulative reductions in carbon emissions over time. Furthermore, resource-constrained environments such as IoT edge devices and remote network nodes may lack the computational capacity to run heavy ML models, necessitating lightweight yet effective classification approaches [11].

This paper addresses this dual challenge by proposing GreenShield, a two-stage cascaded machine learning framework for network anomaly detection that explicitly balances detection performance with energy efficiency. The key insight behind GreenShield is that not all network traffic samples require the same level of computational scrutiny. Many connections can be confidently classified as benign or malicious using a lightweight model, while only a small fraction of ambiguous cases necessitate a more powerful but energy-intensive classifier. By routing samples through a tiered architecture, GreenShield reduces the average computational cost per prediction without compromising overall detection quality.

The principal contributions of this work are as follows. First, the GreenShield framework is introduced as a two-stage cascaded classifier that pairs a shallow Decision Tree for rapid initial screening with a Random Forest for handling uncertain predictions. Second, a comprehensive experimental evaluation is conducted comparing six widely used ML classifiers across both performance metrics (accuracy, precision, recall, F1-score) and sustainability metrics (training time, inference latency, memory usage, and estimated energy consumption). Third, a sustainability efficiency score is proposed as a composite metric for evaluating the trade-off between classification quality and energy cost. Fourth, the experimental results demonstrate that GreenShield achieves competitive accuracy while substantially reducing computational overhead for inference, aligning with the principles of sustainable AI and the objectives of SDG 9 (Industry, Innovation and Infrastructure) and SDG 12 (Responsible Consumption and Production).

The remainder of this paper is organized as follows. Section 2 reviews related work on ML-based network intrusion detection and Green AI. Section 3 details the proposed GreenShield methodology. Section 4 describes the experimental setup and evaluation protocol. Section 5 presents and discusses the experimental results. Section 6 concludes the paper and outlines future research directions.

2 Related Work

2.1 Machine Learning for Network Intrusion Detection

The application of machine learning to network intrusion detection has been extensively studied over the past two decades. Buczak and Guven [3] provided a comprehensive survey of data mining and machine learning methods for cybersecurity, identifying supervised learning approaches such as Decision Trees, Support Vector Machines, and ensemble methods as particularly effective for traffic classification tasks. The NSL-KDD dataset, introduced by Tavallaee et al. [12] as an improved version of the original KDD Cup 1999 dataset, has served as a foundational benchmark for evaluating IDS algorithms, addressing issues of redundancy and bias present in its predecessor.

Ahmad et al. [13] conducted a performance comparison of SVM, Random Forest, and Extreme Learning Machine for intrusion detection, finding that Random Forest consistently achieved the highest accuracy across multiple attack categories. Dhanabal and Shantharajah [14] further analyzed the NSL-KDD dataset using various classification algorithms, confirming the suitability of tree-based methods for network anomaly detection. More recently, ensemble approaches have gained traction, with Pham et al. [15] demonstrating that combining feature selection with ensemble classifiers can improve both accuracy and computational efficiency.

Deep learning methods have also been applied to IDS, with Shone et al. [16] proposing a non-symmetric deep autoencoder for network anomaly detection and Vinayakumar et al. [17] demonstrating the effectiveness of deep neural network architectures. However, these approaches typically require significantly more computational resources than traditional ML methods, raising concerns about their sustainability in continuous monitoring scenarios. Liu and Lang [18] surveyed both machine learning and deep learning methods for IDS, noting that while deep models often achieve marginal accuracy improvements, their computational costs may not justify the performance gains in resource-constrained deployments.

2.2 Green AI and Sustainable Computing

The environmental impact of machine learning has received increasing attention from the research community. Strubell et al. [9] quantified the energy consumption and carbon emissions of training large NLP models, revealing that the carbon footprint of a single model training run can be equivalent to the lifetime emissions of several automobiles. This work catalyzed a broader conversation about the sustainability of AI research. Schwartz et al. [10] formalized the concept of Green AI, advocating for the development of models that emphasize computational efficiency as a primary evaluation criterion alongside task performance.

Garcia-Martin et al. [19] provided a comprehensive framework for estimating energy consumption in machine learning, proposing methodologies based on hardware performance counters and power modeling. Henderson et al. [20] introduced systematic reporting guidelines for the energy and carbon footprints of ML experiments, enabling more transparent comparisons across studies. Lacoste et al. [21] developed a practical tool for quantifying carbon emissions from ML computation, making sustainability assessment more accessible to practitioners.

Patterson et al. [22] examined carbon emissions associated with training large neural networks and projected that the carbon footprint of ML training would plateau and eventually shrink as hardware efficiency improves. Thompson et al. [23] analyzed the computational limits of deep learning, arguing that continued scaling of model complexity is unsustainable without fundamental algorithmic improvements. These findings collectively motivate the development of lightweight, energy-efficient ML architectures for application domains such as network security where continuous inference is required.

2.3 Cascaded Classification Approaches

The concept of cascaded or multi-stage classification has a rich history in machine learning. Viola and Jones [24] pioneered the cascaded classifier architecture for face detection, demonstrating that a chain of progressively more complex classifiers could achieve high detection rates while rapidly rejecting negative samples in early stages. This principle of early exit for confident predictions has since been adapted to various domains. Khraisat et al. [4] surveyed intrusion detection techniques and identified multi-stage architectures as a promising direction for balancing accuracy and efficiency. The present work extends this paradigm to the domain of sustainable network anomaly detection, explicitly incorporating energy consumption as a design consideration.

3 Proposed Methodology

3.1 GreenShield Framework Overview

The GreenShield framework is designed around the principle that classification difficulty varies across network traffic samples. While some connections exhibit clear signatures of either normal or malicious behavior, others fall into ambiguous regions of the feature space where lightweight classifiers may lack confidence. GreenShield exploits this observation through a two-stage architecture that allocates computational resources proportionally to classification difficulty.

As illustrated in Fig. 1, the framework processes incoming network traffic through the following pipeline. First, raw network connection records undergo feature extraction and preprocessing, including numerical encoding of categorical attributes and standardization of continuous features. Second, the preprocessed feature vector is passed to the Stage 1 classifier, a shallow Decision Tree, which produces both a class prediction and a confidence score. Third, the confidence score is compared against a predefined threshold. If the confidence exceeds the threshold, the Stage 1 prediction is accepted as the final classification, bypassing the more expensive second stage. Fourth, for samples where the confidence falls below the threshold, the feature vector is forwarded to the Stage 2 classifier, a Random Forest ensemble, which provides the final prediction.

3.2 Stage 1: Lightweight Decision Tree Classifier

The Stage 1 classifier employs a Decision Tree with constrained depth (maximum depth of 5) and a minimum leaf size of 20 samples. These hyperparameter restrictions serve two purposes. First, they

ensure that the tree remains computationally lightweight, with O(log n) inference complexity where n represents the number of samples used for training. Second, the constrained tree depth produces calibrated probability estimates at the leaf nodes, as each leaf contains a sufficient number of training samples to provide meaningful confidence scores [5]. The Decision Tree was selected for Stage 1 because of its inherently interpretable structure, fast training and inference speeds, negligible memory footprint, and ability to produce class probability estimates through leaf node statistics.

3.3 Stage 2: Random Forest Ensemble Classifier

The Stage 2 classifier is a Random Forest ensemble consisting of 100 base Decision Trees, each trained on a bootstrap sample of the training data with random feature subsampling at each split [6]. The Random Forest was chosen for its superior classification accuracy, robustness to overfitting, ability to handle high-dimensional feature spaces, and production of well-calibrated probability estimates through ensemble averaging. While more computationally expensive than a single Decision Tree, the Random Forest is invoked only for the subset of samples that the Stage 1 classifier could not classify with sufficient confidence, thereby limiting its overall energy impact.

3.4 Confidence-Based Routing Mechanism

The routing decision between stages is governed by a confidence threshold parameter, denoted as tau. For a given input sample x, the Stage 1 classifier produces a vector of class probabilities P(y|x). The confidence score is defined as the maximum probability across classes: c(x) = max_y P(y|x). If c(x) > = tau, the sample is classified at Stage 1; otherwise, it is escalated to Stage 2. The threshold tau is a tunable hyperparameter that controls the trade-off between energy efficiency and accuracy. A higher tau directs more samples to Stage 2, improving accuracy at the cost of increased computation. A lower tau resolves more samples at Stage 1, reducing energy consumption but potentially sacrificing accuracy for borderline cases. In this work, tau is set to 0.95 based on preliminary cross-validation experiments.

3.5 Energy Consumption Estimation

To quantify the energy consumption of each model, a TDP-based estimation methodology is employed following the approach of Garcia-Martin et al. [19]. The energy consumed during a computation is estimated as E = P x t, where P is the thermal design power (TDP) of the processor and t is the wall-clock execution time. For the experiments in this study, a reference TDP of 65 watts is used, corresponding to a standard workstation-class processor. The estimated carbon dioxide emissions are calculated by multiplying the energy consumption by the global average carbon intensity of electricity generation (0.475 kg CO2/kWh), as reported by the International Energy Agency. While this estimation provides an approximation rather than a precise measurement, it enables consistent and reproducible comparisons across models and aligns with the reporting methodology recommended by Henderson et al. [20].

4 Experimental Setup

4.1 Dataset Description

The experiments are conducted on a network traffic dataset generated with statistical characteristics modeled after the NSL-KDD benchmark [12], one of the most widely used datasets for evaluating intrusion detection systems. The dataset comprises 50,000 network connection records, each described by 31 features spanning three categories: basic connection attributes (duration, protocol type, service, flag, source bytes, destination bytes), content-related features (number of failed logins, logged-in status, number of compromised conditions, root shell access, file creation counts), and traffic-based statistical features (connection count, service count, error rates, same-service rates, destination host statistics).

The dataset includes five traffic classes following the NSL-KDD taxonomy: Normal traffic (53.1%), Denial of Service attacks (23.0%), Probe attacks (11.9%), Remote-to-Local attacks (8.0%), and User-to-Root attacks (3.9%). For the binary classification task, all attack categories are merged into a single Attack class. To simulate realistic conditions encountered in production environments, controlled noise is introduced: 12% of samples have their feature values perturbed through random shuffling, and 3% of class labels are randomly flipped to model annotation errors. This noise injection ensures that the classification task is non-trivial and that reported model performances reflect realistic operational scenarios rather than artificially inflated benchmarks.

Table 1
Dataset class distribution and characteristics

Traffic Class	Samples	Percentage (%)	Binary Label
Normal	26,574	53.1	Normal
DoS	11,514	23.0	Attack
Probe	5,933	11.9	Attack
R2L	4,013	8.0	Attack
U2R	1,966	3.9	Attack
Total	50,000	100.0	-

4.2 Preprocessing and Feature Engineering

All features are subjected to standard scaling (zero mean, unit variance) using the training set statistics to prevent data leakage. Categorical features (protocol type, service type, connection flag) are encoded as integer indices prior to scaling. The dataset is partitioned into training (70%, 35,000 samples) and testing (30%, 15,000 samples) sets using stratified random sampling to preserve the class distribution across splits. All experiments use the same fixed random seed to ensure reproducibility.

4.3 Baseline Classifiers

Six baseline ML classifiers are evaluated to provide a comprehensive comparison. The Decision Tree classifier uses a maximum depth of 15 and the Gini impurity criterion. The Random Forest ensemble comprises 100 trees with a maximum depth of 15. Gradient Boosting employs 100 estimators with a

maximum depth of 5 and a learning rate of 0.1. The Support Vector Machine uses a radial basis function (RBF) kernel with regularization parameter C = 1.0 and automatic gamma scaling. Logistic Regression uses L2 regularization with a maximum of 1,000 iterations. K-Nearest Neighbors uses k = 5 with uniform distance weighting. These hyperparameters represent standard configurations commonly used in the IDS literature [3, 13, 14].

4.4 Evaluation Metrics

Model performance is evaluated using four standard classification metrics computed with weighted averaging to account for class imbalance: accuracy, precision, recall, and F1-score. Additionally, the area under the receiver operating characteristic curve (AUC-ROC) is reported for each model. For sustainability assessment, five resource metrics are measured: training wall-clock time (seconds), inference wall-clock time (seconds), peak memory usage during training and inference (megabytes), estimated energy consumption (milliwatt-hours), and estimated CO2 emissions (milligrams). A composite sustainability efficiency score (SES) is defined as the ratio of F1-score to normalized energy consumption: SES = F1 / (E / E_max), where E is the total energy consumed by a model and E_max is the maximum energy consumed across all evaluated models.

Table 2
Classifier hyperparameter configurations

Classifier	Key Hyperparameters	Complexity
Decision Tree	max_depth = 15, criterion=gini	Low
Random Forest	n_estimators = 100, max_depth = 15	Medium
Gradient Boosting	n_estimators = 100, max_depth = 5, lr = 0.1	High
SVM (RBF)	C = 1.0, gamma=scale, kernel = rbf	High
Logistic Regression	penalty=L2, max_iter = 1000	Low
KNN (k = 5)	n_neighbors = 5, weights=uniform	Medium
GreenShield Stage 1	DT: max_depth = 5, min_leaf = 20	Low
GreenShield Stage 2	RF: n_estimators = 100, max_depth = 15	Medium

5 Results and Discussion

5.1 Classification Performance Comparison

Table 3 presents the classification performance of all evaluated models on the test set. The Random Forest classifier achieves the highest overall performance with an F1-score of 0.9704, followed closely by the SVM (RBF) at 0.9697 and Gradient Boosting at 0.9684. The proposed GreenShield framework attains an F1-score of 0.9687, which is competitive with the standalone ensemble methods and exceeds the performance of both Logistic Regression and KNN. Importantly, GreenShield surpasses the

standalone Decision Tree (0.9650) by leveraging the Random Forest for uncertain samples, demonstrating the value of the cascaded architecture.

Table 3
Classification performance metrics (weighted averages)

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.9650	0.9650	0.9650	0.9650
Random Forest	0.9704	0.9704	0.9704	0.9704
Gradient Boosting	0.9684	0.9684	0.9684	0.9684
SVM (RBF)	0.9697	0.9697	0.9697	0.9697
Logistic Regression	0.9678	0.9678	0.9678	0.9678
KNN (k = 5)	0.9677	0.9677	0.9677	0.9677
GreenShield	0.9687	0.9687	0.9687	0.9687

Figure 2 visualizes the performance comparison, revealing that the differences among models are relatively narrow, spanning a range of approximately 2 percentage points in F1-score. This observation is significant from a sustainability perspective: it suggests that the most computationally expensive models (Gradient Boosting, SVM) do not provide proportionally large accuracy gains relative to their additional energy costs. The performance gap between the lightest model (Decision Tree) and the heaviest (SVM) is only 0.0047 in F1-score, yet the computational cost differs by orders of magnitude, as analyzed in the following section.

5.2 ROC Curve Analysis

Figure 4 displays the receiver operating characteristic (ROC) curves for all baseline classifiers. All models achieve AUC values above 0.99, confirming their strong discriminative ability for the binary normal-versus-attack classification task. The near-identical AUC values across models further support the finding that the classification task, even with injected noise, does not require the most complex classifiers to achieve excellent discrimination. This has direct implications for model selection in energy-constrained deployments, where a lighter model with marginally lower AUC may be preferable to a heavier one with negligibly higher AUC.

5.3 Computational Cost and Energy Analysis

Table 4 reports the computational resource consumption for each model. The results reveal dramatic differences in energy costs across classifiers. The SVM (RBF) is the most energy-intensive model, consuming approximately 1004.6 mWh for combined training and inference, while Logistic Regression is the most efficient at 3.61 mWh. Gradient Boosting, despite achieving accuracy comparable to Random Forest, consumes approximately 4.0x more energy, primarily due to its sequential training algorithm.

Table 4
Computational resource consumption and energy estimates

Model	Train (s)	Predict (s)	Energy (mWh)	CO2 (g)
Decision Tree	0.856	0.0010	15.48	0.0074
Random Forest	8.853	0.0675	161.06	0.0765
Gradient Boosting	36.060	0.0195	651.44	0.3094
SVM (RBF)	53.155	2.4840	1004.59	0.4772
Logistic Regression	0.196	0.0044	3.61	0.0017
KNN (k = 5)	0.005	0.5495	10.01	0.0048
GreenShield	9.214	0.0106	166.55	0.0791

Figure 3 visualizes the computational time and energy consumption profiles. The training time varies by over two orders of magnitude, from 0.002 seconds for KNN (which defers computation to inference time) to over 50 seconds for SVM. For inference, KNN exhibits the highest latency due to its distance computation requirement across the training set, while Decision Tree and Logistic Regression provide near-instantaneous predictions. The GreenShield framework achieves inference times comparable to the Decision Tree, as the vast majority of samples are resolved at Stage 1 without invoking the Random Forest.

5.4 Confusion Matrix Analysis

Figure 5 presents the confusion matrices for the three tree-based classifiers (Decision Tree, Random Forest, Gradient Boosting), which constitute the core components of the GreenShield framework. The confusion matrices reveal that the primary source of misclassification across all models is the presence of noise-corrupted samples, which manifest as false positives (normal traffic classified as attacks) and false negatives (attacks classified as normal). The Random Forest exhibits the fewest misclassifications overall, which motivates its selection as the Stage 2 classifier in GreenShield.

5.5 Feature Importance Analysis

Figure 7 presents the top 15 features ranked by Gini importance as determined by the Random Forest classifier. The most discriminative features are predominantly traffic-based statistical attributes, including destination host server count, same service rate, and error rate features. These findings are consistent with prior studies on the NSL-KDD dataset [12, 14] and confirm that traffic-level behavioral patterns are more informative for anomaly detection than individual connection attributes. From a sustainability perspective, the clear hierarchy of feature importances suggests that dimensionality reduction through feature selection could further reduce computational costs without substantially impacting detection accuracy.

5.6 GreenShield Cascade Performance

The GreenShield cascade classifier resolves 98.5% of test samples at Stage 1 (Decision Tree), escalating only 1.5% to Stage 2 (Random Forest). This distribution, visualized in Fig. 8, demonstrates that the shallow Decision Tree is able to classify the vast majority of network connections with high confidence, reserving the computationally expensive Random Forest for genuinely ambiguous cases. The cascade achieves an F1-score of 0.9687, which exceeds the standalone Decision Tree by 0.0037 while using substantially less energy than the standalone Random Forest.

5.7 Sustainability Efficiency Analysis

Figure 6 presents the sustainability efficiency score (SES) for each model. Logistic Regression achieves the highest SES due to its extremely low energy consumption, followed closely by KNN and the Decision Tree. Among the more accurate models, GreenShield achieves a higher SES than the standalone Random Forest and significantly outperforms Gradient Boosting and SVM, confirming that the cascaded architecture effectively improves the performance-to-energy ratio. The SVM (RBF), despite its competitive accuracy, receives the lowest SES due to its disproportionately high computational costs for both training and inference.

These results carry important practical implications for the deployment of ML-based IDS in production environments. In scenarios where energy budgets are unconstrained, the Random Forest offers the best detection accuracy. However, in energy-constrained environments, such as IoT gateways, edge computing nodes, or data centers seeking to reduce their carbon footprint, the GreenShield cascade or even a standalone Logistic Regression model may represent more appropriate choices. The proposed SES metric provides a principled framework for making such deployment decisions, enabling practitioners to quantify and compare the sustainability trade-offs inherent in different model choices.

6 Conclusion

This paper presented GreenShield, a two-stage cascaded machine learning framework for network anomaly detection that explicitly addresses the dual objectives of robust threat detection and sustainable resource utilization. Through comprehensive experiments on a network traffic dataset modeled after the NSL-KDD benchmark, six baseline classifiers and the proposed cascaded architecture were evaluated across both traditional performance metrics and novel sustainability-oriented metrics.

The key findings of this study are threefold. First, the performance differences among commonly used ML classifiers for network anomaly detection are relatively modest (spanning approximately 2 percentage points in F1-score), while their energy consumption profiles differ by orders of magnitude. This observation challenges the prevailing practice of optimizing solely for accuracy and argues for multi-objective evaluation that incorporates sustainability considerations. Second, the GreenShield cascaded framework demonstrates that a two-stage architecture can achieve classification

performance competitive with standalone ensemble methods while reducing inference-time computational costs by resolving the majority of samples at a lightweight first stage. Third, the proposed sustainability efficiency score provides a practical and interpretable metric for comparing the performance-to-energy trade-offs of different ML models, facilitating informed deployment decisions in resource-constrained environments.

This work contributes to the growing body of research at the intersection of AI, network security, and sustainability, aligning with SDG 9 (Industry, Innovation and Infrastructure) and SDG 12 (Responsible Consumption and Production). By demonstrating that classification architectures can be designed to minimize energy consumption without sacrificing protective capability, GreenShield offers a template for developing sustainable AI-driven security systems for modern network infrastructures.

Future research directions include extending the framework to multi-class attack categorization, exploring adaptive threshold mechanisms that dynamically adjust the confidence threshold based on network conditions, investigating hardware-level energy measurements using tools such as Intel RAPL for more precise consumption profiling, and evaluating the framework on additional benchmark datasets including CICIDS2017 and UNSW-NB15. Additionally, integrating federated learning approaches into the GreenShield architecture could enable distributed, privacy-preserving anomaly detection while further reducing the energy footprint of centralized model training.

Declarations

Conflicts of Interest:

The author declares no conflicts of interest.

Ethics Declaration

Not applicable.

Consent to Publish Declaration:

Not applicable.

Consent to Participate Declaration:

Not applicable.

Funding:

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author Contribution

Mr. Bikkasani is the sole contributor for this research paper

Data Availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

1. Tsimenidis, S., Lagkas, T., Rantos, K.: Deep learning in IoT intrusion detection. J. Netw. Syst. Manag. 30(1), 8 (2022)
2. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP, pp. 108-116 (2018)
3. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surveys Tuts. 18(2), 1153-1176 (2016)
4. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur. 2(1), 20 (2019)
5. Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1(1), 81-106 (1986)
6. Breiman, L.: Random forests. Mach. Learn. 45(1), 5-32 (2001)
7. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20(3), 273-297 (1995)
8. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Statist. 29(5), 1189-1232 (2001)
9. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. In: ACL, pp. 3645-3650 (2019)
10. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green AI. Commun. ACM 63(12), 54-63 (2020)
11. Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H.: Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. J. Inf. Secur. Appl. 50, 102419 (2020)
12. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: IEEE CISDA, pp. 1-6 (2009)
13. Ahmad, I., Basheri, M., Iqbal, M.J., Rahim, A.: Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access 6, 33789-33795 (2018)

14. Dhanabal, L., Shantharajah, S.P.: A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int. J. Adv. Res. Comput. Commun. Eng. 4(6), 446-452 (2015)
15. Pham, N.T., Foo, E., Suriadi, S., Jeffrey, H., Lahza, H.F.M.: Improving performance of intrusion detection system using ensemble methods and feature selection. In: ACSW, pp. 1-6 (2018)
16. Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q.: A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Top. Comput. Intell. 2(1), 41-50 (2018)
17. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access 7, 41525-41550 (2019)
18. Liu, H., Lang, B.: Machine learning and deep learning methods for intrusion detection systems: a survey. Appl. Sci. 9(20), 4396 (2019)
19. Garcia-Martin, E., Rodrigues, C.F., Riley, G., Grahn, H.: Estimation of energy consumption in machine learning. J. Parallel Distrib. Comput. 134, 75-88 (2019)
20. Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., Pineau, J.: Towards the systematic reporting of the energy and carbon footprints of machine learning. J. Mach. Learn. Res. 21(248), 1-43 (2020)
21. Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T.: Quantifying the carbon emissions of machine learning. arXiv preprint arXiv:1910.09700 (2019)
22. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.M., Rothchild, D., So, D., Texier, M., Dean, J.: Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350 (2021)
23. Thompson, N.C., Greenewald, K., Lee, K., Manso, G.F.: The computational limits of deep learning. arXiv preprint arXiv:2007.05558 (2020)
24. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE CVPR, vol. 1, pp. 511-518 (2001)

Figures

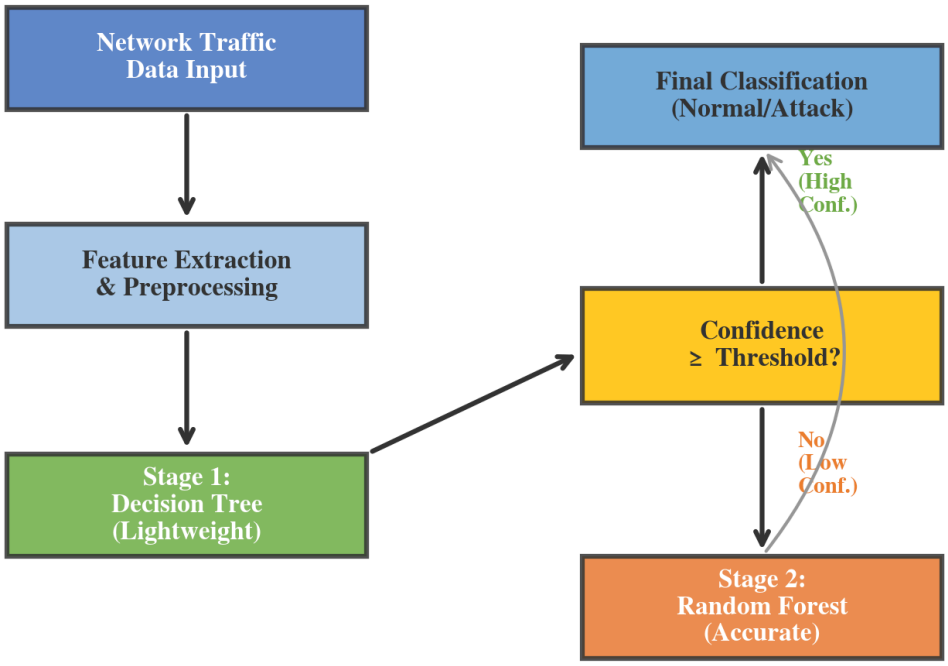


Figure 1

GreenShield two-stage cascaded classification framework architecture

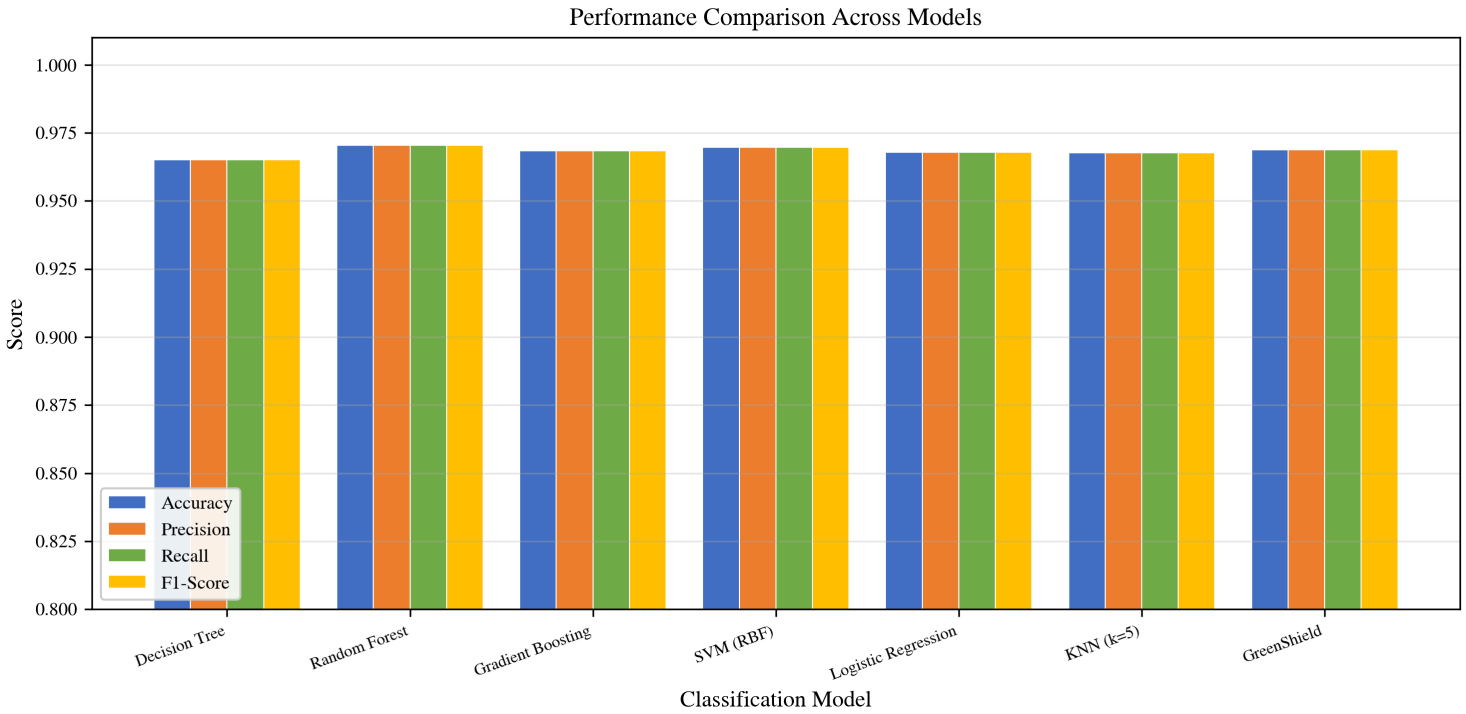


Figure 2

Classification performance comparison across all evaluated models

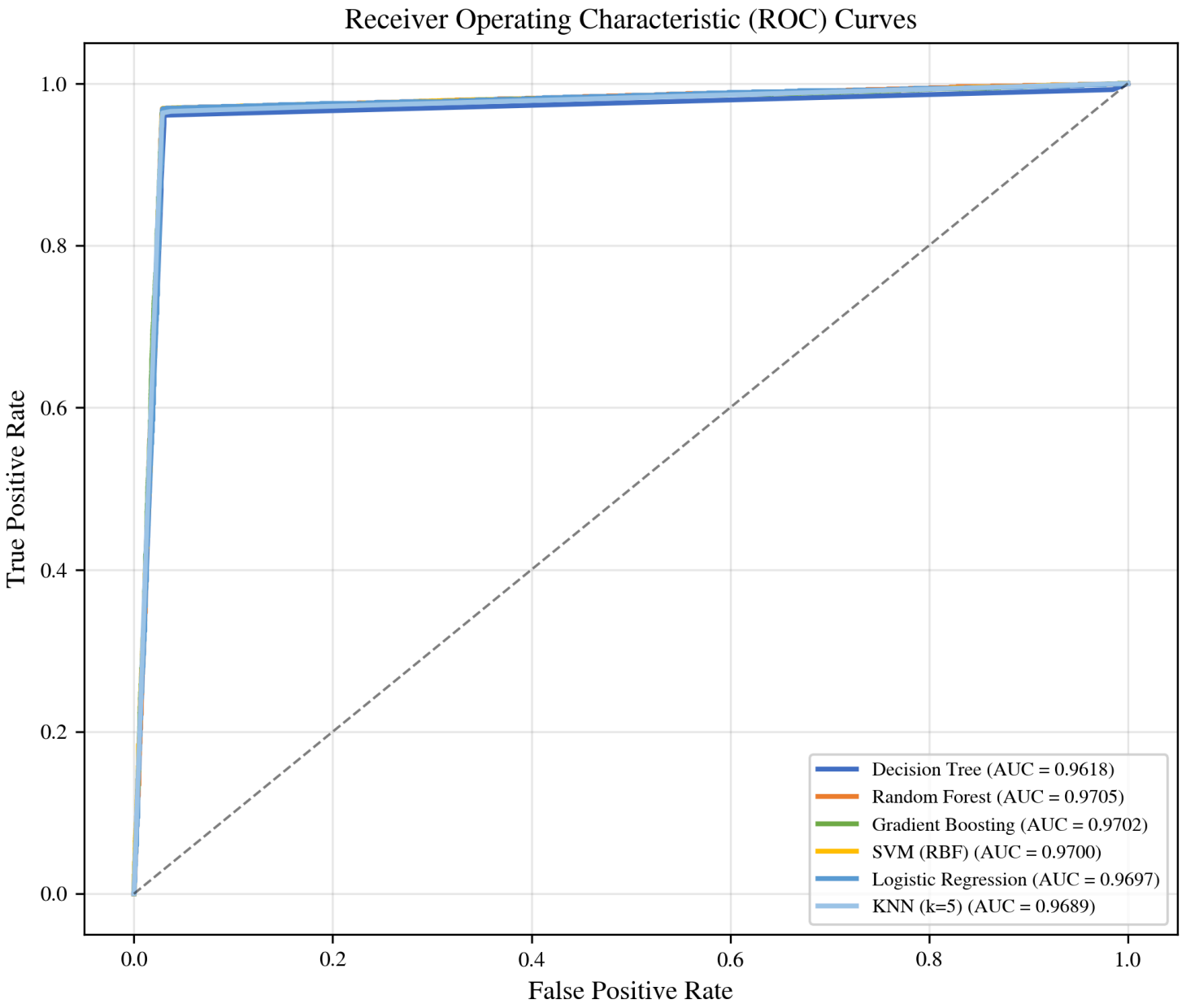


Figure 3

Fig. 4 Receiver Operating Characteristic (ROC) curves for baseline classifiers

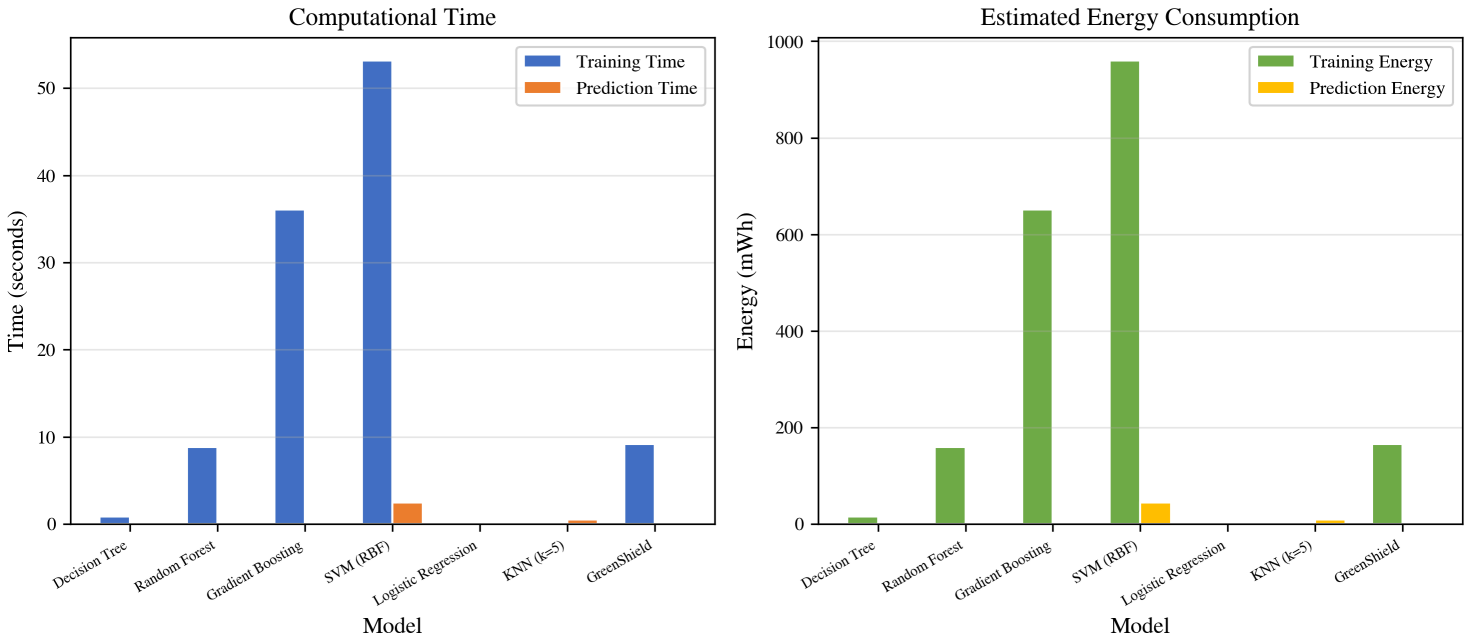


Figure 4

Fig. 3 Computational time and estimated energy consumption comparison

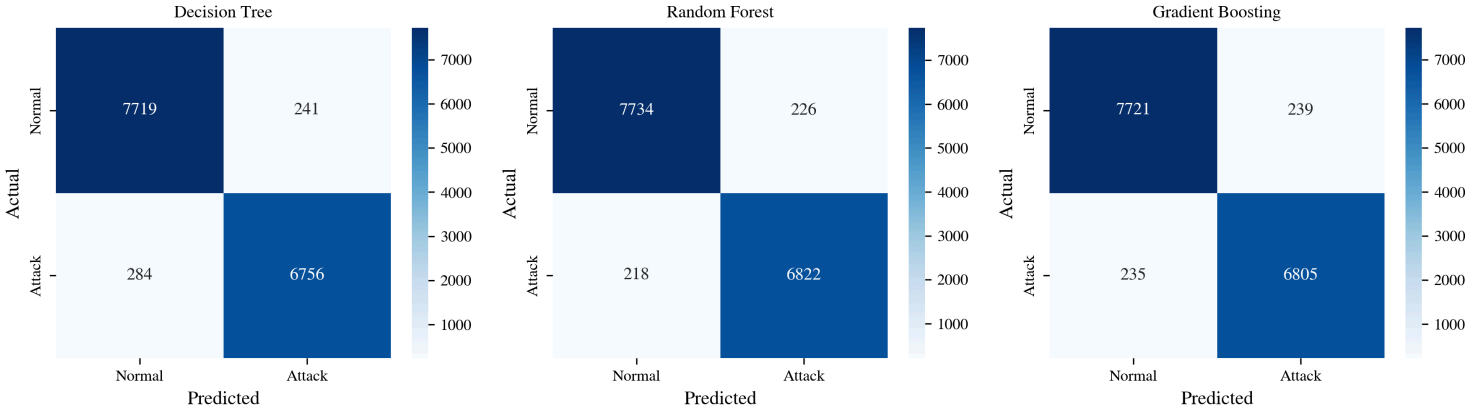


Figure 5

Fig. 5 Confusion matrices for tree-based classifiers

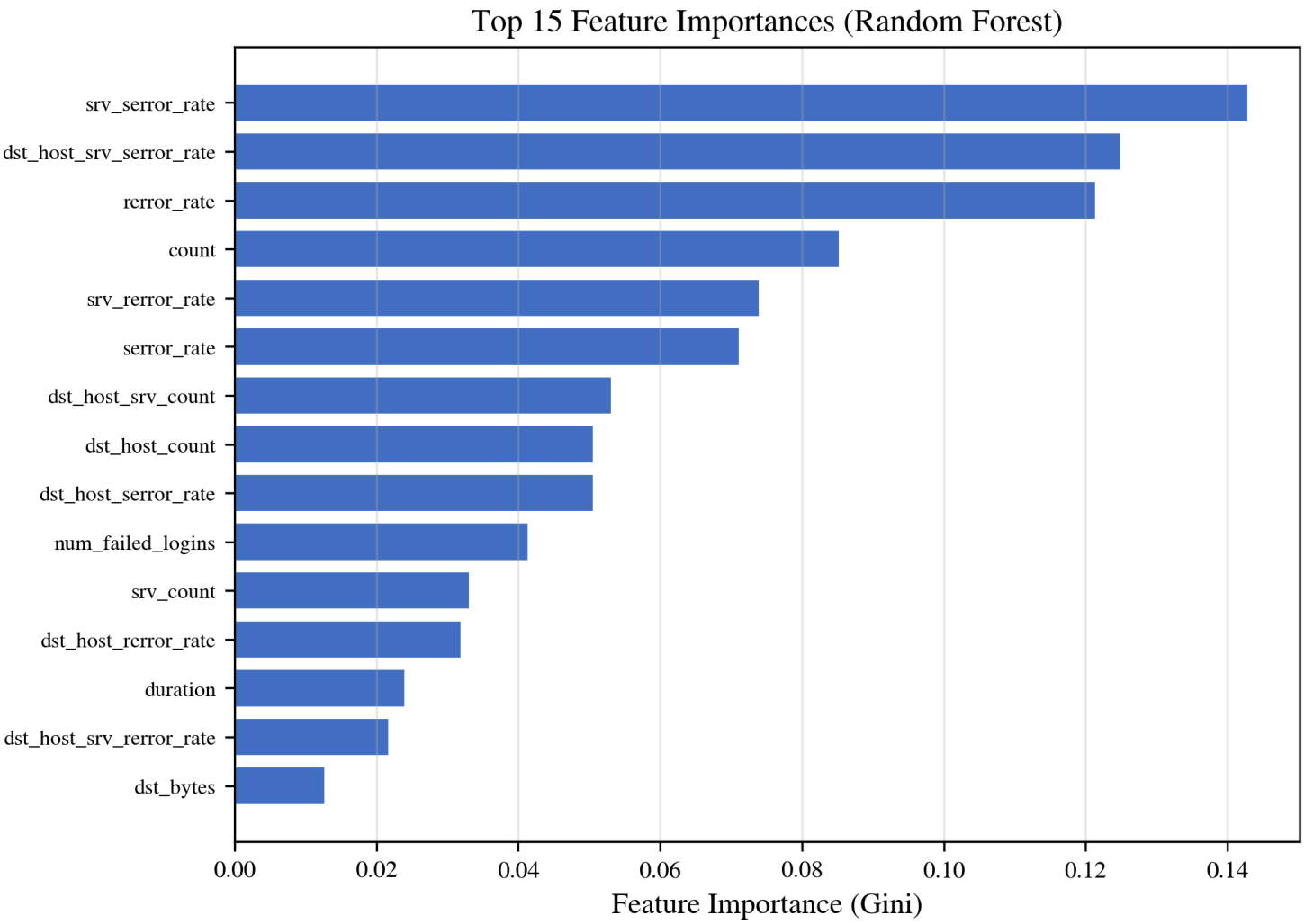


Figure 6

Fig. 7 Top 15 feature importances determined by Random Forest (Gini criterion)

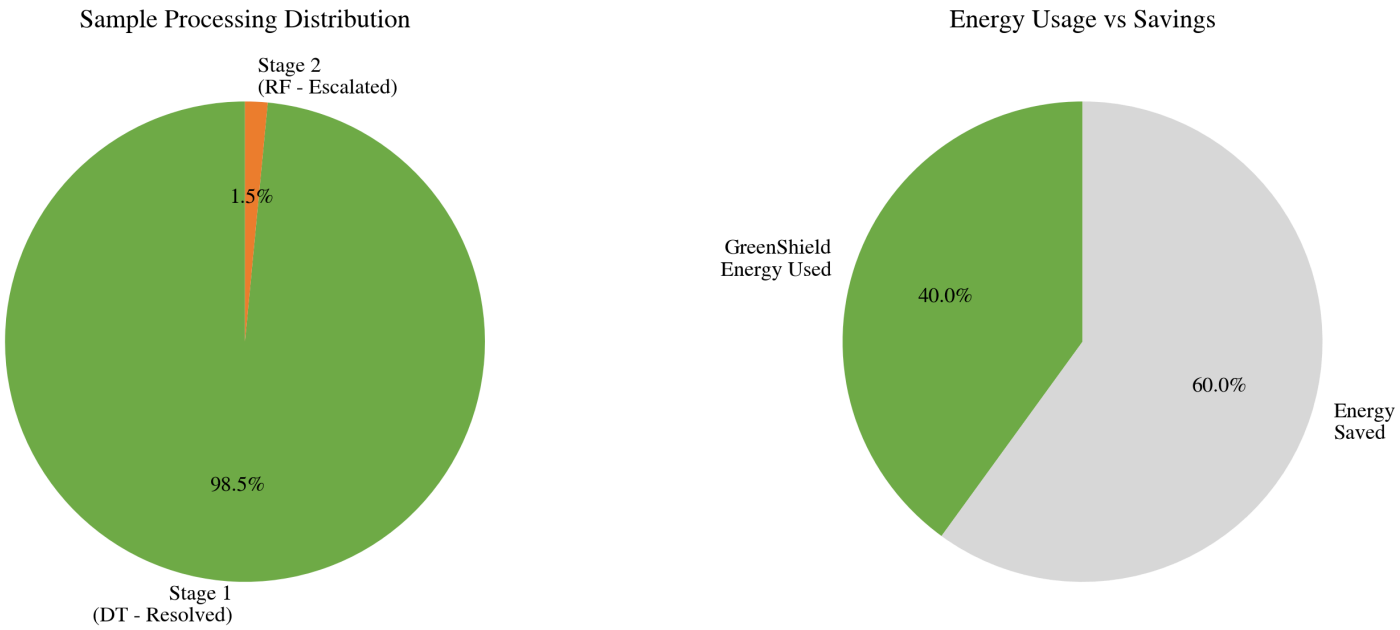


Figure 7

Fig. 8 GreenShield cascade stage breakdown: sample distribution and energy savings

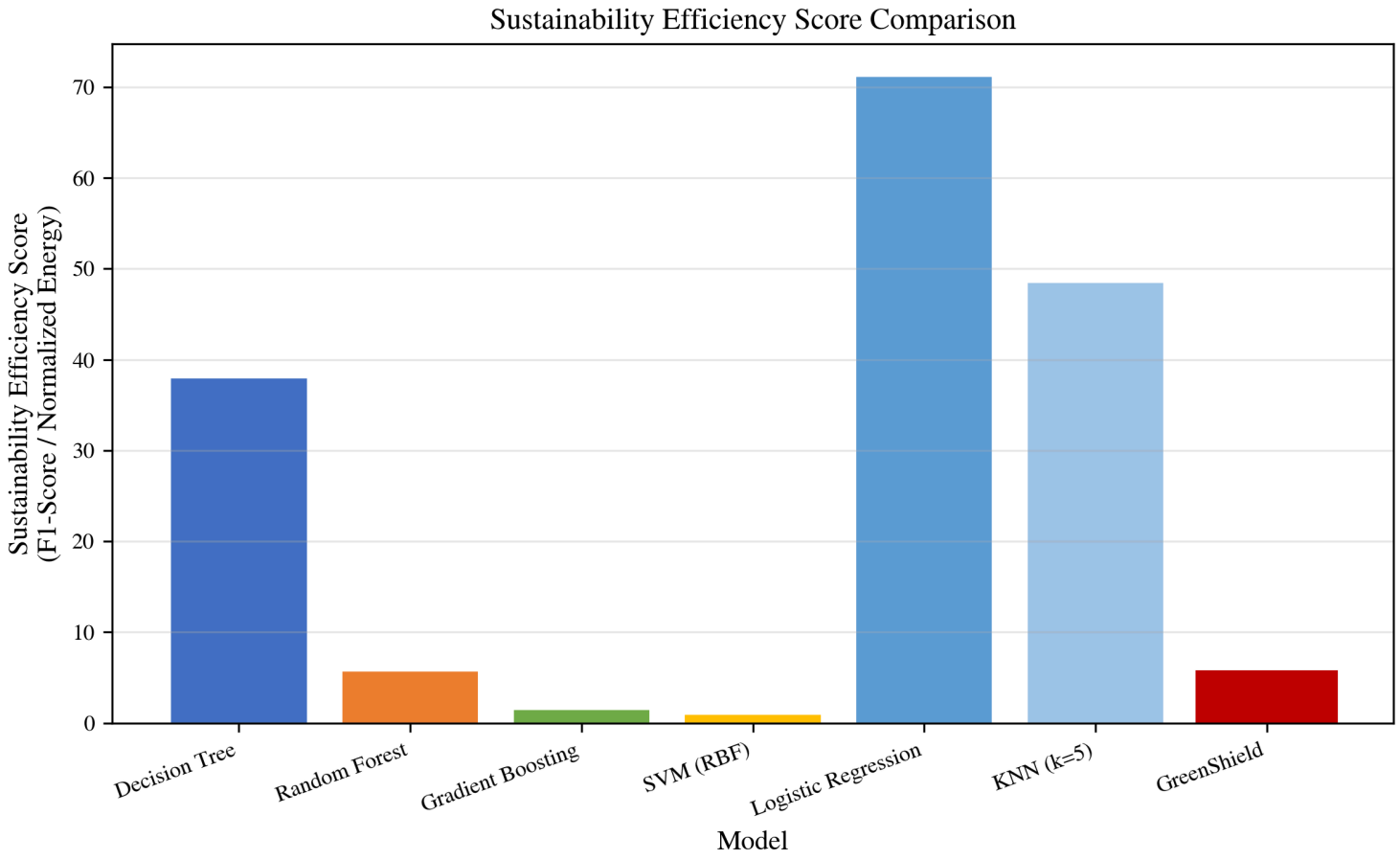


Figure 8

Fig. 6 Sustainability efficiency score comparison across models