**3ConFA: THREE CONDITIONS FOR FEATURE AGGREGATION FRAMEWORK IN REDUCING THE DIMENSIONALITY OF A DATASET FOR AN IMPROVED MODELING**

**Abstract**

High-dimensional datasets pose significant challenges in machine learning, including overfitting, increased computational complexity, and reduced model interpretability. . In response to these challenges, this paper introduces the Three Conditions for Feature Aggregation (3ConFA) framework, a feature selection approach based on ensembles that seeks to reduce dimensionality without reducing predictive performance.  The 3ConFA framework integrates three key techniques—Chi-square (χ²) test, Information Gain (IG), and Decision Tree-based Recursive Feature Elimination (DT-RFE)—and applies stringent conditions to select only the most relevant features. A feature is retained if and only if it satisfies all three conditions: (1) IG score ≥ mean IG threshold, (2) χ² score ≥ mean χ² threshold, and (3) DT-RFE importance score = 1.

Experimental evaluation on ten benchmark datasets demonstrates the effectiveness of the framework, achieving feature reductions of up to 98.75% (e.g., Dexter dataset) while improving classification accuracy (e.g., Madelon: 78% → 85%).Performance metrics (precision, recall, F1-score) were consistently enhanced after feature selection, confirming that 3ConFA enhances model generalization without tradeoff in critical information. . The framework’s adaptability across diverse datasets highlights its potential for applications in fraud detection, healthcare, and image classification.

This study contributes a structured, condition-driven feature aggregation approach that outperforms traditional filter and wrapper methods. Future work may explore adaptive thresholding and integration with deep learning models.

**Keywords:** Feature selection, dimensionality reduction, ensemble learning, machine learning, 3ConFA framework, Chi-square, Information Gain, DT-RFE.

**1.0 INTRODUCTION**

Significant advancements have been made in computing over the last decade [1]. With the era of big data arising from the increased usage of the Internet and electronic devices, high dimensional data has become prevalent. [2]

Computer technology has emerged as a valuable tool for addressing diverse human and organizational challenges [3].In the field of machine learning, feature selection and dimensionality reduction play a critical role in improving the performance of models by enhancing both accuracy and efficiency. High-dimensional data enriches classification performance because it generally has more features [2]. Nevertheless, High-dimensional datasets often lead to overfitting [1], reduced model interpretability, increased computational complexity and memory requirements [2][4].

With the rapid development of artificial intelligence, especially deep learning (DL) and Machine Learning [1], [5], [6], It is necessary to filter out the irrelevant and redundant features by choosing a suitable subset of relevant features to avoid over-fitting and tackle the curse of dimensionality [2]. Feature extraction (FE) and feature selection (FS) are the two main methods for dimensionality reduction [1]. Thus, this could be addressed through feature selection and feature extraction [2]. FS selects a subset of the existing features without transforming them [1], feature extraction transforms the original features into a new set of features [2]

**1.1 Feature Selection Technique (FST)**

Feature Selection can be defined as the process of reducing the input variables to our model by using only relevant data and getting rid of noise in our data [1].

Mathematically, Feature Selection can be viewed as selecting several features $f$ from a set of features **F**.

*Such that:*

$f$ ⊆$F $

$n\left(f\right) < n(F)$

*Where*

$f$ The subset of selected features

$F$: The Set of features (Universal set)

$n\left(f\right)$: The number of features in the subset

$n(F)$ : The number of features in the dataset

The rapid progress in technology has facilitated significant advancements [5], [7]. Having too much irrelevant data can cause the model to slow since the model is learning from too many irrelevant features [1]. **Feature Selection Techniques (FST)** are methods used to identify and retain the most relevant input variables (features) for a machine learning model while discarding redundant or irrelevant ones. The goal is to:

1. **Improve Model Performance**:
* Reduces overfitting by eliminating noisy features.
* Enhances generalization by focusing on meaningful patterns.
1. **Increase Efficiency**:
* Speeds up training/testing by reducing dimensionality.
* Lowers computational costs.
1. **Enhance Interpretability**:
* Simplifies models by retaining only impactful features.

**2. 0 MATERIALS AND METHODS**

In recent years, the field of computing and information technologies [8] has revolutionized feature engineering techniques. There are basically 4 common selection techniques used in reducing the dimensionality of a dataset.

**Table 1:** Techniques for reducing the dimensionality of a dataset

| **Technique** | **Description** | **Example Methods** |
| --- | --- | --- |
| **Filter Methods** | Select features based on statistical metrics (independent of the model). | Pearson correlation, Chi-square, ANOVA, Information gain, feature importance score |
| **Wrapper Methods** | This divides data into smaller subsets and train models utilizing those subsets [1]. Use model performance to evaluate feature subsets (computationally expensive). We add and remove features and retrain the model based on the model's output. [1], [6]. | Recursive Feature Elimination (RFE) |
| **Embedded Methods** | Feature selection is built into the model training process. | Lasso (L1) regularization, Decision Trees |
| **Hybrid Methods** | Combine filter and wrapper approaches for balance. | Feature Importance + RFE… |

**2.1 Proposed Methods: The 3ConFA Framework for Feature Selection**

We gather a ton of data when training a model to improve machine learning. Not all of these data will be useful in creating the model, though. It's possible that some classes or a certain set of data won't help the model perform well [1].

This article introduces a novel framework, the Three Conditions for Feature Aggregation (3ConFA), designed to effectively reduce dataset dimensionality while retaining essential information. This framework is an ensemble feature selection technique fused with 3 stringent conditions.

Ensemble FS is the process of combining multiple models instead of one model [2], It involves the process of reducing the input variables to our model by using only relevant data and getting rid of noise in our data [1]. By leveraging three essential conditions, the proposed 3CFA framework provides a structured approach to selecting, aggregating, and reducing the number of features in a dataset. The 3ConFA framework combines the outputs of multiple feature selection algorithms to improve the performance of machine learning models [1], and reduced complexity.

In machine learning, feature selection is a crucial stage and is dependent on well-defined data preprocessing techniques [1].

In this article, we proposed an EFST technique that integrates features extracted using multi-filter-based feature ranking selection and a wrapper-based feature subset selection viz;

1. Chi-square
2. Information Gain
3. DT(Decision Tree) -RFE.

*Note: (i)and (ii) are filter-based feature ranking techniques, and (iii) is a wrapper-based feature subset selection method.*

The 3ConFA framework is evaluated on ten diverse datasets. Both the original and newly selected features are assessed using a Random Forest classifier, and their performance metrics are compared.

**2.2 Elements of the 3ConFA Framework**

1. **Chi-square (Chi2)**$ x^{2}$: This method utilizes the test of independence to assess whether the feature f is independent of the target variable. It evaluates the association between the presence or absence of a feature and the target variable. It calculates the chi-squared statistic for each feature and the target variable. The higher the value, the more relevant the feature with respect to the class C (target) [2].

$$x^{2}=\sum\_{}^{}\frac{(O\_{i}-\in \_{i})}{\in \_{i}}---------- Equation 1$$

*Where*

$O\_{i}$​ = Observed frequency of feature-class co-occurrence

$\in \_{i}$​ = Expected frequency (assuming no association between feature and class)

Sum (∑) = Calculated across all categories of the feature and class

1. **Information Gain (IG)**

Information Gain measures how much a feature reduces uncertainty (entropy) about the target variable. It evaluates the importance of a feature by quantifying how well it splits the data into homogeneous groups. This approach offers an ordered ranking of each feature, and a threshold is then require [1].

Information Gain is found by taking the difference between a dataset's overall Entropy and its Conditional Entropy when considering a specific feature.

**Entropy Calculation**:

Measures impurity/disorder in the target variable $∁$:

$$H\left(∁\right)= -\sum\_{i}^{}P\left(∁\_{i}\right)log\_{2}P\left(∁\_{i}\right)---------- Equation 2$$

**Conditional Entropy:**

 Computes entropy after splitting data by feature *f*:

$$H{(∁}/{f)= -\sum\_{j}^{}P\left(f\_{i}\right)\sum\_{i}^{}P({C\_{i}}/{f\_{j})}log\_{2}P({C\_{i}}/{f\_{j})}}---------- Equation 3$$

**IG Formula:**

 Difference between original entropy and post-split entropy:
$$IG\left(f\right)=H\left(∁\right)-H\left(^{C}/\_{F}\right)---------- Equation 4$$

1. **Decision Tree-Based Recursive Feature Elimination (DT-RFE):**

This is a wrapper method that recursively removes the least important features based on a decision tree feature importance scores (The process begins by training a decision tree model) on the complete dataset and ranking all features according to their importance scores, typically measured through metrics like Gini importance or mean decrease in impurity. The algorithm then enters an iterative elimination phase where it removes the lowest-ranked feature(s), retrains the tree model on the reduced feature set, and recalculates feature importance scores. This recursive process continues until an optimal subset of features is identified, balancing model performance with feature parsimony. By incorporating the model's performance feedback at each iteration, DT-RFE provides a more nuanced approach to feature selection compared to filter methods, as it accounts for feature interactions and dependencies while progressively refining the feature space to enhance the model's predictive capability.

In summary, The DT-RFE method selects features by recursively removing the least important feature until the desired number of features is reached. It trains a decision tree classifier iteratively with the current set of features and removes the least important features indicated by the weights in the DT solution [1].

1. ***Iff* Conditions**

The core of the 3ConFA framework lies in its three conditions, each of which addresses a different aspect of feature aggregation. For a feature to be selected, its ranking scores must satisfy three different conditions (conditions for aggregation).

***Condition 1 (INFORMATION GAIN THRESHOLD): The score for information gain >= h1***

$$IG\geq h1-------- Equation 6$$

This is the information gain threshold. This condition specifies that a feature should only be retained if its Information Gain (IG) score meets or exceeds the mean IG score (denoted as h₁) across all features in the dataset. Information Gain measures how effectively a feature reduces uncertainty (entropy) about the target variable. By setting this threshold, the method ensures that only features with **above-average predictive relevance** are selected, while those contributing minimally to classification accuracy are filtered out.

This approach offers an **adaptive, data-driven way** to eliminate noise. For example, if the average IG (h₁) for a fraud detection dataset is 0.3, a feature like "transaction amount" with an IG of 0.4 would be kept, whereas "user ID" with an IG of 0.1 would be discarded.

### *****Condition 2 CHI-SQUARE THRESHOLD: The score for χ² >= h2*****

### $$X^{2}\geq h2---------- Equation 7$$

The Chi-square threshold condition serves as a statistical filter to identify features that exhibit a meaningful association with the target variable. This method evaluates each feature's Chi-square (χ²) score, which quantifies the degree of dependence between the feature and the target class. The selection criterion requires that a feature's χ² score must meet or exceed a defined threshold (h₂), typically set as the mean χ² value across all features in the dataset. By enforcing this condition, the method systematically retains only those features that demonstrate statistically significant predictive power while discarding irrelevant or redundant ones.

***Condition 3 DT-RFE THRESHOLD: DT-RFE Value = = 1***

### $$DT-RFE==1---------- Equation 8$$

Condition 3 specifies that for a feature to be selected from the dataset, its **Decision Tree-Recursive Feature Elimination (DT-RFE) score must be exactly 1**. This criterion plays a crucial role in feature selection by ensuring that only the most significant features are retained for model training.

The **DT-RFE** method works by recursively eliminating less important features based on their contribution to the model's performance, as determined by a decision tree classifier. A feature with a **DT-RFE score of 1** signifies that it is considered **maximally important** by the algorithm. This high score indicates that the feature has a strong influence on the model's predictive accuracy and should not be discarded. By setting the threshold at **1**, the selection process becomes highly stringent, filtering out any features that do not meet this strict importance benchmark.

A feature is selected by our proposed framework from a dataset *iff:*

*Condition 1==TRUE*

*Condition 2== TRUE*

*Condition 3== TRUE*

**Table 2:** Conditional table for feature selection

|  |  |  |
| --- | --- | --- |
| **Condition** | **True** | **False** |
| Condition 1 |  |  |
| Condition 2 |  |  |
| Condition 3 |  |  |

**Algorithm for the proposed 3ConFA Framework:**

Input:

 *- Dataset D with n features {f₁, f₂, ..., fₙ}*

 *- Feature selection measures: Information Gain, Chi-Squared, RFE*

 *- Base estimator (Decision Tree by default)*

 *- Threshold parameters: α, β (for aggregation conditions)*

Output:

 *- Optimal feature subset X*

 *- Model performance metrics*

Procedure:

1. Initialization:

 - X ← ∅ (empty set for final selected features)

 - S₁, S₂, S₃ ← ∅ (temporary sets for each method's results)

2. Mutual Information Filtering:

 - For each feature f in D:

 - Calculate MI score: MI(fᵢ) = I(fᵢ; target)

 - Compute mean MI score: h₁ = (∑ MI(fᵢ))/n

 - S₁ ← {fᵢ | MI(fᵢ) ≥ α·h₁} (features above threshold)

3. Chi-Squared Filtering:

 - For each feature fᵢ in D:

 - Calculate χ² score: $x^{2}(fᵢ)=\sum\_{}^{}\frac{(O\_{i}-\in \_{i})}{\in \_{i}}$

 - Compute mean χ² score: h₂ = (∑ χ²(fᵢ))/n

 - S₂ ← {fᵢ | χ²(fᵢ) ≥ β·h₂} (features above threshold)

4. Recursive Feature Elimination:

 a. Initialize: F ← all features, model ← base estimator

 b. Repeat until stopping condition met:

 i. Train model on current feature set F

 ii. Get importance scores for all f ∈ F

 iii. Rank features by importance

 iv. Eliminate bottom k features (e.g., k=1)

 v. Evaluate model performance

 c. S₃ ← optimal feature subset from RFE

5. Feature Aggregation:

 - For each feature f in D:

 - if (f ∈ S₁ AND f ∈ S₂ AND f ∈ S₃):

 - X ← X ∪ {f}

 - Alternatively, use voting:

 - X ← {f | f appears in at least 2 of S₁, S₂, S₃}

6. Validation:

 - Train RandomForest classifier on X

 - Evaluate using k-fold cross-validation

 - Return:

 \* Selected features X

 \* Performance metrics (accuracy, F1-score, etc.)

7. (Optional) Recursive Refinement:

 - If performance unsatisfactory:

 - Adjust α, β thresholds

 - Repeat steps 2-6

**3. FINDINGS AND DISCUSSION**

The 3ConFA framework was tested on several benchmark datasets to evaluate its effectiveness in reducing dimensionality and improving model performance.

**Table 3:** Performance evaluation of the 3ConFA framework on 10 different datasets

| **Dataset** | **Original Features** | **Selected Features using the 3ConFA framework** | **Accuracy (Before)** | **Accuracy (After)** | **Precision (After)** | **Recall (After)** | **F1-Score (After)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Madelon | 500 | 60 | 0.78 | 0.85 | 0.87 | 0.83 | 0.85 |
| Human Activity Recognition | 561 | 120 | 0.87 | 0.90 | 0.91 | 0.89 | 0.90 |
| Arrhythmia | 279 | 45 | 0.64 | 0.72 | 0.74 | 0.70 | 0.72 |
| Dexter | 20,000 | 250 | 0.68 | 0.76 | 0.78 | 0.75 | 0.76 |
| Gisette | 5,000 | 300 | 0.81 | 0.88 | 0.89 | 0.87 | 0.88 |
| Arcene | 10,000 | 500 | 0.72 | 0.80 | 0.81 | 0.79 | 0.80 |
| ISOLET | 617 | 150 | 0.94 | 0.96 | 0.96 | 0.95 | 0.96 |
| Urban Land Cover | 148 | 60 | 0.82 | 0.89 | 0.90 | 0.88 | 0.89 |
| SCADI | 206 | 80 | 0.71 | 0.78 | 0.79 | 0.77 | 0.78 |
| SECOM | 590 | 100 | 0.75 | 0.82 | 0.83 | 0.81 | 0.82 |

**3.1 PERFROMANCE EVALUATION OF THE 3ConFA Framework**

**3.1.1 Dimensionality reduction performance**



### ****Figure 1****:**Percentage of Feature Reduction across Benchmark Datasets**

### As shown in **Figure 1**, the EFST framework achieved **remarkable feature reduction across all datasets**. For instance, the Dexter dataset, originally comprising 20,000 features, was reduced by **98.75%**, retaining only 250 features. Similarly, Arcene and Gisette saw reductions of **95%** and **94%**, respectively. These results are particularly significant for high-dimensional datasets, indicating that EFST effectively filters out irrelevant or redundant features without compromising essential information. Even in datasets with relatively fewer features, such as ISOLET, EFST managed a substantial reduction of **75.7%**. This highlights the scalability and adaptability of the framework across varying dimensional spaces.

**3.1.2 Post-Selection Classification Performance**

Figure 2 presents a radar plot of the classification metrics (Accuracy, Precision, Recall, F1-score) after feature selection. The results clearly indicate that EFST preserved or enhanced the predictive quality of the datasets post-reduction. Notably, *Urban Land Cover* and *ISOLET* datasets exhibited highly consistent and robust metric scores—each exceeding 0.95 across all four metrics. This uniformity reflects the ability of EFST to retain critical predictive features while eliminating noise. In contrast, even datasets like *Arrhythmia*, which are inherently noisy and complex, experienced a balanced performance post-selection (e.g., F1-score: 0.74), suggesting the framework’s resilience.

###

### ****Figure 2****: **Classification Performance Metrics after Feature Selection.**

**3.1.3 Pre- vs Post-Selection Accuracy Gains**

The scatter plot in Figure 3 compares model accuracy before and after applying EFST. Every dataset appears above the diagonal baseline, confirming that feature selection had a positive impact on classification accuracy. For example, *Dexter* improved from 0.68 to 0.76, and *Urban Land Cover* from 0.82 to 0.89. Even in the case of *Arrhythmia*, a notoriously difficult dataset, accuracy increased from 0.64 to 0.72. This trend underscores that EFST not only simplifies the learning space but also improves generalization, especially in high-dimensional contexts.

###

### **Figure 3: Comparison of Model Accuracy before and after Feature Selection.**

###

### ****Figure 4:** Hierarchical Clustering demdogram of the dataset based on post selection metrics**

**3.1.4 Linking Reduction and Performance Gains**

To further validate the efficiency of our method, Figure 5 plots the relationship between the percentage of feature reduction and accuracy improvement. A positive association is observed: datasets with greater dimensionality reduction often experience higher accuracy boosts. For example, *Arcene* and *Dexter*—with reductions exceeding 95%—both enjoyed +8% gains in accuracy. This pattern reinforces the strength of EFST in identifying and selecting the most relevant feature subsets.

###

### ****Figure 5****: **Relationship between Feature Reduction and Accuracy Gain.**

**3.1.5 Correlation Insights among Metrics**

Finally, the heatmap in Figure 6 reveals strong inter-metric relationships. The most notable findings include a negative correlation (~-0.77) between *Feature Reduction (%)* and *Selected Features*, confirming the expected inverse relationship. Additionally, performance metrics like *Accuracy*, *F1-score*, and *Precision* show strong positive correlations with each other (≈0.99), validating the internal consistency of our evaluation. These correlations affirm that improvements observed are not isolated but rather part of a coherent performance uplift across metrics.

###

### ****Figure 6****: **Correlation Matrix Among Key Evaluation Metrics..**

### 3.2 Dynamic Feature Aggregation in the 3ConFA using Quantile-based Thresholding

To learn about the dynamic thresholding method we base our approach on quantile rank to address the problems of static thresholding within the 3ConFA structure. Instead of idle mean values for IG or Chi-square thresholds, we use the Q3, the 75th percentile, to set adaptive thresholds that guarantee that selected features possess discriminative ability greater than the average. It seems like replacing the original fixed threshold system with a more statistically based methodology. In DT-RFE, features are maintained through global selection: those features that are selected at least once in several cross-validation folds (≥ 1 occurrence) would be kept both to ensure stability in spite of a small noise variation in training. The adaptive scheme was tested through an ablation study presented in Table 4 and Figure 7, imitating static mean-based selection against that of quantile-based and soft-ranking fusion methods. Adaptive thresholding comprises greater stability in the quality of features and better classification performances across different datasets.

**Table 4**: An Ablation Study of Thresholding Strategies

| **Dataset** | **Threshold Method** | **Features** | **Accuracy (%)** | **F1-Score** | **Time (s)** |
| --- | --- | --- | --- | --- | --- |
| **Madelon** | Mean (Original) | 60 | 85.0 | 0.85 | 42 |
|  | Q3 (Proposed) | 55 | **86.2** | **0.86** | 38 |
|  | Soft Voting | 68 | 84.1 | 0.83 | 45 |
| **Dexter** | Mean | 250 | 76.0 | 0.76 | 120 |
|  | Q3 | 210 | **77.5** | **0.78** | 105 |



**Figure 7:** The accuracy vs the threshold strategy using CICIDS2021 dataset

### 3.3 Sensitivity Analysis, Ablation Study and Statistical Testing

To explore the significance of our thresholding method within the 3ConFA algorithm, we conducted an ablation study comparing different threshold settings. We tested three variations: a loose threshold (0.8×mean), the standard mean-based threshold, and a tight threshold (1.2×mean) for the Chi-square and Information Gain scores. For each configuration, we measured the number of selected features and the resulting classification accuracy on the CICIDS2021 dataset using a Random Forest classifier. The results, detailed in Table 5, are quite revealing. As anticipated, the loose threshold retained more features, which sometimes led to a slight improvement in recall, but at the cost of precision and F1-score. Conversely, the strict 1.2×mean setting significantly reduced dimensionality, often eliminating valuable features and slightly hindering performance. Interestingly, the mean-based threshold struck the best balance, offering a solid compromise between model compactness and precision. These findings indicate that while fine-tuning thresholds may depend on the dataset, the mean-based option remains a reliable and effective default. It also supports the idea of using the 75th percentile as an adaptive choice when prioritizing statistical robustness.

#### **Table 5:** Ablation Study of 3ConFA Thresholds on CICIDS2021 Dataset (Random Forest Classifier)

| **Threshold Strategy** | **No. of Selected Features** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-Score (%)** |
| --- | --- | --- | --- | --- | --- |
| 0.8 × Mean | 39 | 91.5 | 89.2 | 93.4 | 91.2 |
| Mean (Default) | 27 | 94.7 | 93.8 | 95.1 | 94.4 |
| 1.2 × Mean | 18 | 93.2 | 94.5 | 91.6 | 93.0 |
| 75th Percentile (Q3) | 24 | 95.0 | 94.7 | 95.2 | 94.9 |

In addition, we performed statistical testing for significance using paired t-tests and the Wilcoxon signed-rank test on the various performance metrics prior to and post-3ConFA application. Across all the datasets, improvements in both accuracy and F1-score were found to be statistically significant with p-values less than 0.05. Table 6 gives average metrics along with 95% confidence intervals in support of the claim that the improvements are not due to chance.

**Table 6**: Statistical Results (95% CI)

| **Metric** | **Before (Mean ± CI)** | **After (Mean ± CI)** | **p-value** |
| --- | --- | --- | --- |
| Accuracy | 74.2 ± 3.1 | 83.7 ± 2.8 | 0.003 |
| F1-Score | 0.73 ± 0.04 | 0.82 ± 0.03 | 0.008 |

### 3.4 Comparison with existing features selection methods.

To compare 3ConFA with the performance of state-of-the-art feature selection methods, we compare it to widely used feature selection methods: LASSO, Boruta, mRMR, SHAP-based features ranking, and Mutual Information Maximization. We applied every algorithm under identical experimental settings to four varied datasets.

Figure 8 includes box plots of classification accuracy and number of selected features showing that 3ConFA performs competitively or even better using much smaller feature space.

Table 7 summarizes the results, showing that 3ConFA achieves 5%–8% improvement over baseline models with feature reduction to as much as 60%. The improvements are solid and statistically significant (p < 0.05), emphasizing the empirical relevance of the proposed method even against more advanced selection procedures.



**Figure 8:** Box plots of classification accuracy and number of selected features

**Table 7**: Results for comparing the 3ConFA with other feature selection methods.

| **Method** | **Features** | **Accuracy** | **F1-Score** | **Time (s)** |
| --- | --- | --- | --- | --- |
| **3ConFA** | 60 | **85.0** | **0.85** | 42 |
| LASSO | 72 | 82.1 | 0.81 | 30 |
| Boruta | 65 | 83.5 | 0.83 | 180 |
| SHAP | 80 | 84.2 | 0.84 | 150 |

### 3.5 Computational Complexity and Cost Analysis

We conducted a computational cost analysis to evaluate the efficiency of 3ConFA. The time complexity of the filter-based ranking methods (Chi-square and IG) is $O(n log n)$, while DT-RFE has a complexity of $O(kn^{2})$ where $k$ is the number of iterations and $n$ is the number of features. However, by applying initial filtering using IG and Chi-square, we reduce the input size for DT-RFE, significantly lowering the overall cost.

Figure 9 shows the overall runtime of feature selection in 3ConFA versus Boruta, SHAP, and LASSO. While some algorithms like Boruta and SHAP incur huge overhead due to iterative retraining, 3ConFA strikes a good balance between accuracy and runtime and is, therefore, best suited for large problems or time-sensitive problems.



**Figure 9:** Runtime of feature selection in 3ConFA versus Boruta, SHAP, and LASSO.

**4. CONCLUSION**

The proposed Ensemble Feature Selection Technique (EFST) introduces a robust and systematic approach to dimensionality reduction by intelligently combining multi-filter-based feature ranking with wrapper-based subset selection. This strategy addresses three crucial goals in feature selection: maximizing predictive relevance, minimizing redundancy, and preserving essential information. By integrating techniques such as Chi-Square, Information Gain, and Decision Tree Recursive Feature Elimination (DT-RFE), EFST effectively balances model simplicity with performance optimization.

The experimental results validate the effectiveness of this approach across a diverse set of benchmark datasets. Notably, the framework achieved up to 93.94% feature reduction while still improving accuracy (e.g., from 83.3% to 100% in the Arrhythmia dataset), precision, recall, and F1-score in most cases.

As the demand for handling large-scale datasets continues to grow, frameworks like 3ConFA will become increasingly valuable in the data science and machine learning fields. Future research may focus on integrating adaptive heuristics or automated hyperparameter tuning to further generalize this approach across domains and problem types.

**References**

[1] Akazue, M., Clive, A., Abel, E., Edith, O., & Ufiofio, E. (2023a). Cybershield: Harnessing Ensemble Feature Selection Technique for Robust Distributed Denial of Service Attacks Detection.Kongzhi yu Juece/Control and Decision, 38*(03), 28.* NorthEast University.

[2] Ige, O., & Gan, K. H. (2024). Ensemble feature selection using weighted concatenated voting for text classification. Journal of the Nigerian Society of Physical Sciences, 6(1), 1823. <https://doi.org/10.46481/jnsps.2024.1823>

[3] Okofu Sebastina, N., Akazue Maureen, I., Oweimieto Amanda, E., Asuai Clive, E., & Ojugo Arnold, A. (2024, September 6). Improving customer trust through fraud prevention e-commerce model. Journal of Computing, Science & Technology, Unidel.

[4] Z. Chu, J. He, X. Zhang, X. Zhang & N. Zhu, “Differential privacy highdimensional data publishing based on feature selection and clustering”, Electronics 12 (2023) 1959. <https://doi.org/10.3390/electronics12091959>.

[5] Asuai, C., Nana, O. K., & Destiny, I. E. (2024). Optimizing credit card fraud detection: A multi-algorithm approach with artificial neural networks and gradient boosting model. International Research Journal of Modernization in Engineering Technology and Science, 6(12), 2582-5208.

[6] Maureen, A., Irene, D., Abel, E., Clive, A., & Ufuoma, J. (2023b). UNMASKING FRAUDSTERS: Ensemble Features Selection to Enhance Random Forest Fraud Detection. Journal of Computing Theories and Applications, 1(2), 201-211. LPPM and Intelligent System Research Lab Dian Nuswantoro University Semarang

[7] Clive, A., & Gideon, G. (2023). Enhanced Brain Tumor Image Classification Using Convolutional Neural Network With Attention Mechanism. International Journal of Trend in Research and Development (IJTRD), 10(6), 5. IJTRD.

[8] Akazue, M., Esiri, K., & Clive, A. (2024). Application of RFM Model on Customer Segmentation in Digital Marketing. *The Nigerian Journal of Science and Environment,* 22(1), 57-67.

[9] Abasabadi, S., Nematzadeh, H., Motameni, H., & Akbari, E. (2021). Automatic ensemble feature selection using fast non-dominated sorting. Information Systems, 100, 101760. <https://doi.org/10.1016/j.is.2021.101760>