**Introduction**

Thyroid disorders represent a significant global health burden, affecting hundreds of millions of individuals worldwide, particularly women and the elderly. These disorders—ranging from hypothyroidism and hyperthyroidism to thyroid nodules and cancer—can lead to severe metabolic, cardiovascular, and cognitive complications if undetected or mismanaged. In response to this challenge, machine learning (ML) has increasingly been adopted in clinical research and diagnostics to enhance early detection, risk stratification, and personalized treatment of thyroid diseases. Central to the effectiveness of such ML models is the quality and sequence of data preparation, especially the steps involving feature selection and dataset splitting, which directly impact the model’s accuracy, generalizability, and clinical applicability.

Feature selection, particularly in biomedical datasets, is essential for eliminating irrelevant or redundant predictors, reducing dimensionality, and improving model interpretability. Techniques such as Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), and Decision Tree (DT)-based feature importance are frequently employed in thyroid-related predictive modeling. However, despite their proven utility, improper application especially before splitting the data into training and testing sets can result in data leakage, a phenomenon that leads to overly optimistic performance metrics and reduces reproducibility. This issue is especially critical in healthcare applications, where biased predictions can have real-world consequences on patient safety and treatment decisions.

Recent literature has increasingly raised concerns about this methodological flaw. Studies between 2018 and 2025 have demonstrated how conducting feature selection prior to dataset splitting contaminates the test data and violates the fundamental principle of evaluating models on unseen data. Research institutions and regulatory agencies alike including those involved in medical device approval and digital health standards now emphasize strict validation practices. In thyroid disease research, where datasets often involve imbalanced classes, high-dimensional biomarkers, and small sample sizes, ensuring that feature selection is performed only on training data is vital to producing robust and clinically meaningful models.

This study responds to these concerns by systematically evaluating the proper timing of feature selection methods in the context of thyroid disease prediction. It focuses specifically on how methods like RFE, PCA, and DT-based selection behave when applied before versus after dataset splitting. The goal is to build a framework that avoids data leakage and supports reliable model development using both global datasets and real-world clinical records from thyroid patients. This framework aligns with best practices in medical AI and provides a reproducible guideline for researchers working with diagnostic data.

In doing so, this research contributes not only to methodological clarity but also to policy-oriented practice. It informs clinicians, data scientists, and healthcare AI developers about the technical and ethical implications of premature feature selection. By highlighting the risks of invalid evaluation and offering an operational solution, the study supports the broader objective of creating trustworthy machine learning systems in medicine especially those applied to endocrine and thyroid disorders.

In summary, while the application of ML in thyroid disease prediction is expanding, foundational issues in pipeline design such as the sequence of feature selection and data partitioning remain under-addressed in many studies. This paper investigates this critical point, establishes a validated workflow, and proposes a policy-level guideline grounded in recent evidence. The following Background section explores the clinical significance of thyroid disease, the structure of thyroid diagnostic data, and the evolution of feature selection strategies in medical machine learning.

**Related Work**

Recent literature has reiterated that the sequence of operations in machine learning pipelines is critical. In particular, applying feature selection or dimensionality reduction techniques before the train-test split remains a common source of data leakage—where information from the test set inadvertently guides feature choice—compromising model evaluation and validity

Cross Validated

+5

Stack Overflow

+5

PMC

+5

. Multiple recent reviews and benchmarks confirm that maintaining a strict separation between training and evaluation data is essential for trustworthy model performance assessment.

Princeton researchers in Leakage and the Reproducibility Crisis in ML-based Science (2023) evaluated nearly 300 studies across various fields and found pervasive misuse of data splits—especially leakage during feature selection and preprocessing steps

Wikipedia

+2

reproducible.cs.princeton.edu

+2

ScienceDirect

+2

. Their work underscores that even subtle leaks can inflate reported accuracy, giving a false sense of model robustness, and hindering reproducibility across disciplines.

Neuroimaging provides a concrete example of this pitfall. A 2024 study on connectome-based prediction revealed that pipelines selecting features across full datasets—rather than only within training splits—produced dramatically inflated Pearson’s r values: up to +0.47 in some cases

Cross Validated

+15

Nature

+15

ScienceDirect

+15

. Their results clearly demonstrate how leakage can distort both model evaluation and scientific conclusions when feature selection is implemented prematurely.

Similarly, a Nature publication from 2022 examining deep-learning models found that improper dataset splitting inflated Matthews correlation by 0.07–0.43 (i.e., 5–30% accuracy improvement)

Stack Overflow

+1

Nature

+1

Nature

. This finding highlights that even sophisticated models trained on modern medical imaging data are vulnerable to gross misestimation when leakage is present at any pipeline stage.

From a methodological perspective, an arXiv review (2023) on LASSO‑based feature selection showed how even multi-stage pipelines—which seemingly honor cross-validation protocols—can inadvertently leak information when feature ranking and model training are decoupled across folds

Nature

+15

arXiv

+15

arXiv

+15

. Their audit highlights that textual descriptions often mask implementation errors that reintroduce leakage despite proper theoretical frameworks.

Efforts to formalize leakage types have also advanced. A January 2025 Journal of Big Data article outlines a taxonomy covering “premature featurization” and “group leakage”—where feature selection spans across subject clusters or temporal groups—and reiterates the general principle to confine feature learning to training folds only

arXiv

Wikipedia

. Their taxonomy guides practitioners to systematically identify and avoid leak sources across diverse data modalities.

In practice, recent guidelines for feature stores and data pipelines publish purged or embargoed validation protocols to prevent look‑ahead leakage. Feathr and other frameworks now support point‑in‑time joins to enforce time-safe splits for feature computation, ensuring that feature engineering—including selection and transformation—occurs entirely within the training window

vldb.org

. These advances reflect growing recognition that pipeline automation must enforce isolation of train/test data rigorously.

Together, these works from 2018 onward reinforce a unified message: feature selection must be restricted to training data. Whether in neuroimaging, medical imaging, or business analytics, failure to isolate this step induces data leakage, inflating performance, reducing reproducibility, and misleading scientific or business decisions. The literature is clear—preserving the independence of the test set remains a non‑negotiable standard for credible ML evaluation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Study / Author(s)** | **Method(s) Investigated** | **Key Insight** | **Risk Addressed** | **Citation** |
| 1 | Kuhn & Johnson (2019) | RFE, Filter, Embedded | Feature selection should be inside CV loop and after split. | Data leakage, bias | [1] |
| 2 | James et al. (2021) | General ML pipeline | All selection must occur on training data to ensure proper generalization. | Invalid evaluation | [2] |
| 3 | Guyon & Elisseeff (2003) | RFE (SVM) | Performing FS outside split causes information leakage. | Data leakage | [3] |
| 4 | Ambroise & McLachlan (2002) | Gene selection (biomedical) | High false discovery when feature selection precedes splitting. | Overfitting | [4] |
| 5 | Saeys et al. (2007) | DT, Wrapper, Filter | Emphasized FS must be nested in training to be reliable. | Invalid estimation | [5] |
| 6 | Varma & Simon (2006) | Feature selection with CV | FS before CV results in selection bias and overestimated accuracy. | Overfitting, bias | [6] |
| 7 | Hastie, Tibshirani & Friedman (2009) | PCA, ElasticNet | PCA and other transforms must be trained only on training data. | Data leakage | [7] |
| 8 | Brownlee (2020) | PCA, RFE, Embedded | All transformations and selections must follow splitting. | Reproducibility | [8] |
| 9 | Cawley & Talbot (2010) | Model evaluation bias | Demonstrated inflated accuracy when pre-selection used entire dataset. | Invalid evaluation | [9] |
| 10 | Bommert et al. (2020) | Benchmark of FS methods | Evaluation pipelines that use test data in FS suffer from performance inflation. | Bias, poor generalization | [10] |

**Methodology**

This section presents the complete methodology used to investigate the effect of feature selection timing on thyroid disease prediction using real-world clinical data. The study focuses on the correct application of three popular feature selection techniques—**Recursive Feature Elimination (RFE)**, **Principal Component Analysis (PCA)**, and **Decision Tree-based feature importance**—and evaluates their impact on predictive performance when applied only after dataset splitting. The experimental design includes preprocessing, data splitting, feature selection, model training, and evaluation.

A screenshot of a computer

AI-generated content may be incorrect.

A diagram of a model training

AI-generated content may be incorrect.

**2.1 Dataset and Preprocessing**

The dataset was obtained from **Smart Towers Hospital (Sulaymaniyah, Iraq)**, consisting of thyroid-related patient records with diagnostic labels. Features included demographic variables (e.g., age, gender), clinical measurements (e.g., TSH, T3, T4), and behavioral indicators (e.g., smoking status).

The preprocessing pipeline included:

* **Handling missing values** using **KNN imputation** for numerical features and **mode imputation** for categorical features.
* **Label encoding** of categorical variables.
* **Outlier detection** using **Z-score** thresholds (|Z| > 3).
* **Standardization** using:

Z=X−μσZ = \frac{X - \mu}{\sigma}Z=σX−μ​

where XXX is the feature value, μ\muμ is the mean, and σ\sigmaσ is the standard deviation.

**2.2 Dataset Splitting**

The data was split into **training (80%) and testing (20%) sets** using **stratified sampling** to preserve class distribution. Importantly, **feature selection was applied only to the training set** to prevent data leakage. The learned feature subset or transformation was then applied to the test set for evaluation.

**2.3 Feature Selection Techniques**

Three methods were used for feature selection, each applied **only on the training set**.

**2.3.1 Recursive Feature Elimination (RFE)**

RFE is a wrapper method that recursively removes the least important features based on model coefficients. Given a model fff, feature ranking is computed, and features with the lowest importance are removed iteratively until kkk features remain.

For a linear model (e.g., logistic regression):

y^=σ(β0+∑j=1nβjxj)\hat{y} = \sigma(\beta\_0 + \sum\_{j=1}^{n} \beta\_j x\_j)y^​=σ(β0​+j=1∑n​βj​xj​)

where:

* y^\hat{y}y^​ is the predicted probability
* βj\beta\_jβj​ is the coefficient of feature xjx\_jxj​
* σ(⋅)\sigma(\cdot)σ(⋅) is the sigmoid function

Features are ranked based on the magnitude ∣βj∣|\beta\_j|∣βj​∣, and the smallest coefficients are eliminated recursively.

**2.3.2 Principal Component Analysis (PCA)**

PCA is an unsupervised method that reduces dimensionality by projecting data onto orthogonal components with maximum variance.

Let X∈Rn×pX \in \mathbb{R}^{n \times p}X∈Rn×p be the standardized training matrix. PCA computes eigenvectors and eigenvalues of the covariance matrix:

Σ=1n−1X⊤X\Sigma = \frac{1}{n-1} X^\top XΣ=n−11​X⊤X

Solving:

Σv=λv\Sigma v = \lambda vΣv=λv

where:

* λ\lambdaλ is the eigenvalue (explained variance)
* vvv is the eigenvector (principal component)

The projection of the data onto the first kkk components is:

Z=XVkZ = X V\_kZ=XVk​

where Vk∈Rp×kV\_k \in \mathbb{R}^{p \times k}Vk​∈Rp×k contains the top kkk eigenvectors.

**2.3.3 Decision Tree-based Feature Importance**

A decision tree estimates feature importance using **Gini impurity reduction**. For a node ttt with split on feature jjj, importance is computed as:

ΔIG(j,t)=IG(t)−pLIG(tL)−pRIG(tR)\Delta I\_G(j, t) = I\_G(t) - p\_L I\_G(t\_L) - p\_R I\_G(t\_R)ΔIG​(j,t)=IG​(t)−pL​IG​(tL​)−pR​IG​(tR​)

where:

* IG(t)=1−∑i=1Cpi2I\_G(t) = 1 - \sum\_{i=1}^{C} p\_i^2IG​(t)=1−∑i=1C​pi2​ is the Gini impurity of node ttt
* pLp\_LpL​, pRp\_RpR​ are proportions of samples in the left and right child nodes
* IG(tL)I\_G(t\_L)IG​(tL​), IG(tR)I\_G(t\_R)IG​(tR​) are impurities after the split

The total importance for feature jjj across all nodes is:

Importance(j)=∑t∈TjΔIG(j,t)\text{Importance}(j) = \sum\_{t \in T\_j} \Delta I\_G(j, t)Importance(j)=t∈Tj​∑​ΔIG​(j,t)

where TjT\_jTj​ is the set of nodes where feature jjj is used.

**2.4 Model Training and Evaluation**

After feature selection, the selected features (or PCA components) were used to train the following classifiers on the training set:

* **Logistic Regression (LR)**
* **Support Vector Machine (SVM)**
* **Random Forest (RF)**
* **XGBoost (XGB)**

Model training was done using **10×10 stratified cross-validation**, and performance was evaluated on the **held-out test set** using:

* **Accuracy**: TP+TNTP+TN+FP+FN\frac{TP + TN}{TP + TN + FP + FN}TP+TN+FP+FNTP+TN​
* **Precision**: TPTP+FP\frac{TP}{TP + FP}TP+FPTP​
* **Recall (Sensitivity)**: TPTP+FN\frac{TP}{TP + FN}TP+FNTP​
* **F1-score**: 2⋅Precision⋅RecallPrecision+Recall2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}2⋅Precision+RecallPrecision⋅Recall​
* **AUC-ROC**

**2.5 Experimental Design**

To evaluate the effect of selection timing, two pipelines were compared:

* **Correct pipeline**: Feature selection applied **after** splitting (training data only).
* **Incorrect pipeline**: Feature selection applied **before** splitting (entire dataset).

Performance differences were analyzed to highlight the impact of data leakage. A **Consensus Feature Selection (CFS)** method was also tested, selecting features that appeared in **at least two of the three methods**.