OKU Fen Bilimleri Enstitüsü Dergisi
9(2): 1129-1140, 2026

OKU Journal of The Institute of Science and Technology, 9(2): 1129-1140, 2026

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

# Aykırı Değerlere Karşı Normalleştirme Yöntemlerinin Sağlamlığı ve Makine Öğrenmesi Performansı Üzerindeki Etkileri: Sistematik Bir Simülasyon Çalışması

**Hamza YALÇIN[1*], İbrahim ÖZTÜRK[2]**

[1]Department of Animal Science, Faculty of Agriculture, Harran University, Şanlıurfa
[2] Department of Electrical and Electronics Engineering, Faculty of Engineering, Osmaniye Korkut Ata University, Osmaniye

[1]https://orcid.org/0000-0003-0733-7821
[2]https://orcid.org/0000-0003-3149-0527
*Sorumlu yazar: hyalcin@harran.edu.tr

**Araştırma Makalesi**

***Makale Tarihçesi:***
Geliş tarihi: 22.12.2025
Kabul tarihi:05.02.2026
Online Yayınlanma: 16.03.2026

***Anahtar Kelimeler:***
Makine öğrenimi
Normalleştirme
Özellik ölçeklendirme
Aykırı değerler
Sağlam istatistikler

**ÖZ**

Bu çalışma, veri bozulmasının makine öğrenmesi (ML) sınıflandırma algoritmalarının performansı üzerindeki etkisini sistematik bir biçimde incelemektedir. Araştırmanın odağı; aykırı değer oranları, veri boyutluluğu ve farklı öznitelik ölçekleme teknikleri arasındaki karmaşık etkileşimdir. Bu kapsamda, çok değişkenli lojistik regresyona dayalı kapsamlı bir faktöriyel simülasyon tasarımı oluşturulmuş ve yaygın olarak kullanılan dört ML algoritması (XGBoost, Random Forest, LightGBM ve SVM), sekizi farklı ölçekleme yöntemi ve biri normalizasyon uygulanmamış referans durum olmak üzere toplam dokuz ön-işleme koşulu altında karşılaştırmalı olarak değerlendirilmiştir. Çalışmanın temel amacı; örneklem büyüklüğü, değişken sayısı ve veri bozulma şiddeti arttıkça en yüksek düzeyde sağlamlığı koruyan normalizasyon stratejisinin belirlenmesidir. Bulgular, Medyan/MAD (MD) yönteminin, özellikle aykırı değer oranının yüksek olduğu durumlarda (%50'ye kadar), Z-Score gibi geleneksel yöntemlerde gözlenen performans düşüşünü etkili biçimde telafi ederek tutarlı biçimde üstün bir dayanıklılık sunduğunu açıkça ortaya koymaktadır. Her ne kadar ağaç tabanlı topluluk yöntemleri (XGBoost, Random Forest ve LightGBM) doğaları gereği aykırı değerlere karşı çekirdek tabanlı SVM'ye kıyasla daha toleranslı olsa da SVM'nin MD gibi sağlam bir ölçekleyici ile birlikte kullanılması, yüksek derecede kirlenmiş veri kümelerinde modelin kararlılığını belirgin biçimde artırmaktadır.

# The Robustness of Normalization Methods to Outliers and Their Impact on Machine Learning Performance: A Systematic Simulation Study

**Research Article**

***Article History:***
Received: 22.12.2025
Accepted: 05.02.2026
Published online: 16.03.2026

***Keywords:***
Machine learning
Normalization
Feature scaling
Outliers
Robust statistics

**ABSTRACT**

This research provides a systematic investigation into how data corruption impacts the performance of Machine Learning (ML) classification algorithms. Our focus is on the complex interplay between outlier ratios, data dimensionality, and various feature scaling techniques. We constructed a comprehensive factorial simulation based on multivariate logistic regression, benchmarking four widely-used ML algorithms (XGBoost, Random Forest, LightGBM, and SVM) under nine preprocessing conditions: eight feature-scaling methods and one unnormalized baseline. The primary goal was to identify the normalization strategy that maintains the highest level of robustness as data size, feature count, and corruption severity escalate. The findings clearly indicate that the Median/MAD (MD) method demonstrates consistent robustness, especially when the outlier ratio is substantial (up to 50%), effectively counteracting the significant performance degradation seen with conventional methods like Z-Score. While ensemble tree methods

(XGBoost, Random Forest, LightGBM) naturally tolerate outliers better than the kernel-based SVM, pairing a robust scaler like MD with SVM dramatically improves the latter's stability in highly contaminated datasets. This work emphasizes the necessity of context-aware data preprocessing, offering empirically-grounded recommendations for practitioners seeking to build resilient models in data science applications.

**To Cite**: Yalçın H., Öztürk İ. The Robustness of Normalization Methods to Outliers and Their Impact on Machine Learning Performance: A Systematic Simulation Study. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2026; 9(2): 1129-1140.

## 1. Introduction

Machine learning (ML) algorithms have become indispensable tools for tackling complex pattern recognition and prediction problems across a vast spectrum of applications, from scientific research to industrial operations (Bischl et al., 2021). However, the efficacy of any ML model depends critically on the quality of the training data and the rigor of the preprocessing pipeline (Çetin and Yıldız, 2022). Within this pipeline, two steps stand out as paramount: feature scaling (normalization) and the effective management of outliers.

Feature scaling is the process of transforming variables of different magnitudes into a standard range or distribution, ensuring that each feature contributes equitably to the learning process. This step is particularly vital for distance-based algorithms (e.g., SVM) and those relying on gradient descent optimization (Huang et al., 2023). The literature presents numerous scaling techniques, such as Z-Score (Standardization), Min-Max Scaling, and Pareto Scaling (Singh and Singh, 2022).

The fundamental challenge arises because most conventional methods, including Z-Score and Min-Max, rely on statistics (mean and standard deviation, or min and max values) that are highly susceptible to outliers (Alshdaifat et al., 2021). The presence of these extreme values can severely distort the scaling process (Mishra et al., 2020). For instance, Min-Max scaling is highly sensitive to extreme outliers, which can greatly inflate the feature range and compress normal data variation, causing significant signal loss (scikit-learn developers, 2025).

Outliers, defined as observations that deviate significantly from the general pattern of the data, can profoundly degrade the performance and interpretability of ML models (Dash et al., 2023; Khan et al., 2024). They skew both statistical estimates and model parameters, undermining the model's ability to generalize effectively (Khan et al., 2024). The degree of impact varies significantly depending on the algorithm. Kernel-based classifiers like Support Vector Machines (SVM) are highly sensitive to outliers, whereas ensemble tree methods are inherently more robust (Wyatt et al., 2022). This dichotomy has driven researchers to develop robust feature scaling techniques, notably those based on the Median and Median Absolute Deviation (MAD), which are designed to be resistant to extreme values (Tseng et al., 2025).

Existing research often examines the interaction between scaling and outliers within the confines of a single algorithm or a specific real-world dataset (Kandanaarachchi et al., 2020). However, a systematic, factorial simulation that isolates the effects of key data characteristics—specifically sample size, feature count, and outlier ratio—is crucial for developing universal, empirically-grounded guidelines for preprocessing strategy selection (Li et al., 2023; Glotsos et al., 2025). The core objective of this study is to use a rigorous simulation framework to scientifically compare the effect of various normalization methods and outlier ratios on the performance of diverse ML classification algorithms. Our work offers substantial empirical evidence and practical guidance for data scientists and practitioners, particularly those operating in domains characterized by high data noise (Sujon et al., 2024; Kalinina et al., 2025).

## 2. Material and Methods

This section details the systematic simulation design, the data generation process, and the specific feature scaling and machine learning methods employed in this study. The entire simulation was conducted using Python 3.11 with the scikit-learn, pandas, and numpy libraries to ensure full reproducibility. The computational environment used Python 3.11 and the following key libraries: scikit-learn 1.8.0, NumPy (≥1.24.1), SciPy (≥1.10.0), joblib (≥1.3.0), and pandas (≥1.5.0). XGBoost and LightGBM were implemented using xgboost (v3.1.1) and lightgbm (v4.6.0), respectively, via their scikit-learn–compatible estimator interfaces (xgboost.XGBClassifier and lightgbm.LGBMClassifier).

### 2.1. Factorial Simulation Design

We utilized a comprehensive factorial simulation design to investigate the effects of four key factors on classification performance. This approach allows for a thorough examination of not just the main effects but also the interactions between factors.

- **Sample Size:** The number of observations in each dataset was varied across four levels: 100, 250, 500, and 1000.
- **Feature Count:** The number of predictor variables was set to five levels: 5, 10, 20, 30, and 50.
- **Outlier Ratio:** The proportion of contaminated data points was tested at five levels: 0% (a clean baseline), 5%, 10%, 20%, and 50% (extreme contamination).
- **Normalization Method:** Nine conditions were tested, including eight distinct scaling methods and one un-normalized baseline (Table 1).
- **ML Algorithm:** Four popular and structurally different classification algorithms were used.

This design resulted in a total of 4 x 5 x 5 x 9 x 4 = 3,600 unique scenarios. Each scenario was replicated 20 times with different random seeds to ensure the stability and reliability of the results, leading to a total of 72,000 model training and evaluation runs (Glotsos et al., 2025).

### 2.2. Data Generation and Contamination

To ensure the findings are generalizable, we generated synthetic data for a two-class classification problem.

**Data Generation:** For each scenario, a dataset with a specified sample size and feature count was generated. The predictor variables, $X$, were drawn from a multivariate normal distribution with a mean vector of zeros and a covariance matrix Σ. To introduce a realistic level of multicollinearity, the covariance matrix was structured such that the correlation coefficient between any two variables was fixed at $r = 0.70$. Accordingly, Σ was set to an equicorrelation (compound-symmetry) matrix with unit variances ($\Sigma_{ii} = 1$) and constant off-diagonal correlation ($\Sigma_{ij} = \rho = 0.70$, $i \neq j$). The binary class label, $y$, was then generated using a logistic regression model:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + X\beta)}}$$

where the intercept $\beta_0$ was set to 0 and the coefficient vector β was randomly generated from a uniform distribution $U(-1,1)$ for each replication. All elements of β were generated independently from U(−1, 1) at each replication. This parameterization ensures that the resulting datasets are class-balanced (approximately equal distribution of positive and negative classes). Consequently, the outliers introduced in the subsequent step affect the feature space distribution but do not alter the class prevalence, effectively isolating the impact of feature corruption from the class imbalance problem.

**Outlier Injection:** To simulate real-world data corruption, outliers were injected into the predictor variables. For a given outlier ratio, a corresponding percentage of observations were randomly selected. For each selected observation, a random subset of its features (between 1 and the total number of features) was chosen to be contaminated. The value of each chosen feature was then shifted by a random amount drawn from a uniform distribution between $\pm 5$ and $\pm 10$ standard deviations of that feature's original distribution. The shift direction (positive vs. negative) was selected at random with equal probability. This method ensures that outliers are both extreme and sporadic, mimicking measurement errors or other anomalies (Dash et al., 2023).

### 2.3. Normalization Techniques

The study compares eight different normalization methods that are robust against both traditional and outlier values. Normalization transforms the value ($x$) of each variable ($x_i$) into a new value ($x'$) (Table 1).

**Table 1.** Normalization methods, formulas, and explanations are shown.

| Abbreviation | Scaling Method | Reference | Description |
|---|---|---|---|
| **ZS** | Z-Score | $x' = \frac{x - \mu}{\sigma}$ | Scales the data to a mean of zero and a standard deviation of one. It is highly sensitive to outliers because μ and σ are affected by outliers (Singh and Singh, 2022). |
| **MM** | Min-Max Scaling | $x' = \frac{x - min}{max - min}$ | Scales the data to the range [0, 1]. It is sensitive because the min and max values are affected by outliers (scikit-learn developers, 2025). |
| **MD** | Median/MAD | $x' = \frac{x - medyan}{MAD}$ | It uses the median and median absolute deviation (MAD). It is the most robust method against outliers. |
| **MC** | Centered Around the Mean | $x_i' = x_i - \mu$ | It centers the data around the mean. Preserves variance. Sensitive to outliers. |
| **PS** | Pareto Scaling | $x' = \frac{x - \mu}{\sqrt{\sigma}}$ | Similar to ZS, but uses the square root of the standard deviation in the denominator. Reduces the effect of outliers somewhat compared to ZS. |
| **VSS** | Variable Stability Scaling | $x' = \frac{x - \mu}{\sigma} \cdot \frac{\mu}{\sigma}$ | Aims to stabilize the variance of variables. Sensitive to outliers. |
| **MS** | Maximum Scaling | $x' = \frac{x}{max}$ | Divides the data by the $max$ value. Sensitive to a single outlier distorting the $max$ value. |
| **DS** | Decimal Scaling | $x' = \frac{x}{10^k}$ | Divides the data by $10^k$. Sensitive to outliers. |
| **UN** | Unnormalized | $x' = x$ | Raw data without normalization. A critical reference point for testing the inherent robustness of tree-based algorithms. |

### 2.4. Machine Learning Algorithms and Evaluation

Four structurally different algorithms were chosen to assess the impact of normalization across various model families:

- **Ensemble Tree-Based:** XGBoost, Random Forest, and LightGBM. These are known for their high performance and relative robustness (Choudhury et al., 2024).
- **Kernel-Based:** SVM with a radial basis function (RBF) kernel. This algorithm is known to be powerful but sensitive to feature scaling and outliers (Thakur et al., 2025).

For Random Forest and SVM, we used scikit-learn estimators (Pedregosa et al., 2011). For XGBoost and LightGBM, we used their scikit-learn API wrappers (xgboost.XGBClassifier and lightgbm.LGBMClassifier). Unless explicitly stated, hyperparameters were left at their library defaults; only reproducibility-related settings (e.g., random_state and verbosity/logging controls) were fixed. . While hyperparameter tuning (e.g., optimizing the regularization parameter C in SVM) can enhance model robustness, this study intentionally utilizes default settings to isolate the specific impact of feature scaling techniques. Introducing hyperparameter optimization would introduce a confounding variable, making it difficult to decouple the performance gains attributed to normalization from those achieved through model tuning. Furthermore, given the factorial design comprising 72,000 distinct experimental runs, per-scenario optimization was computationally out of scope.

Model performance was evaluated using 10-fold cross-validation. The primary performance metric was the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), which is well-suited for binary classification problems. Given that the synthetic datasets are class-balanced, ROC-AUC provides a comprehensive measure of separability that is invariant to decision thresholds. Unlike F1-score or Precision-Recall, which require a fixed threshold (e.g., 0.5) that may not be optimal across all 72,000 diverse scenarios, ROC-AUC evaluates the fundamental ranking quality of the model produced by each normalization technique.

## 3. Results

This section presents the key findings from proposed comprehensive simulation study, focusing on how different factors influence the ROC-AUC performance metric.

### 3.1. The Global Impact of Outliers and a Visual Overview

Initially, a global analysis reveals that outliers significantly impede model performance. Across all our tests, the more we contaminated the data with outliers, the worse our models performed. The average ROC-AUC score dropped from a respectable 0.9756 in clean datasets to a much lower 0.8133 when half the data was corrupted. This observation validates the fundamental principle that data quality directly constrains model performance.

To give a more detailed picture, Figure 1 shows a heatmap of the ROC-AUC scores for every single scenario we tested. This provides an effective visual representation of how the different algorithms and normalization methods held up as we systematically increased the outlier ratio and changed the data size.

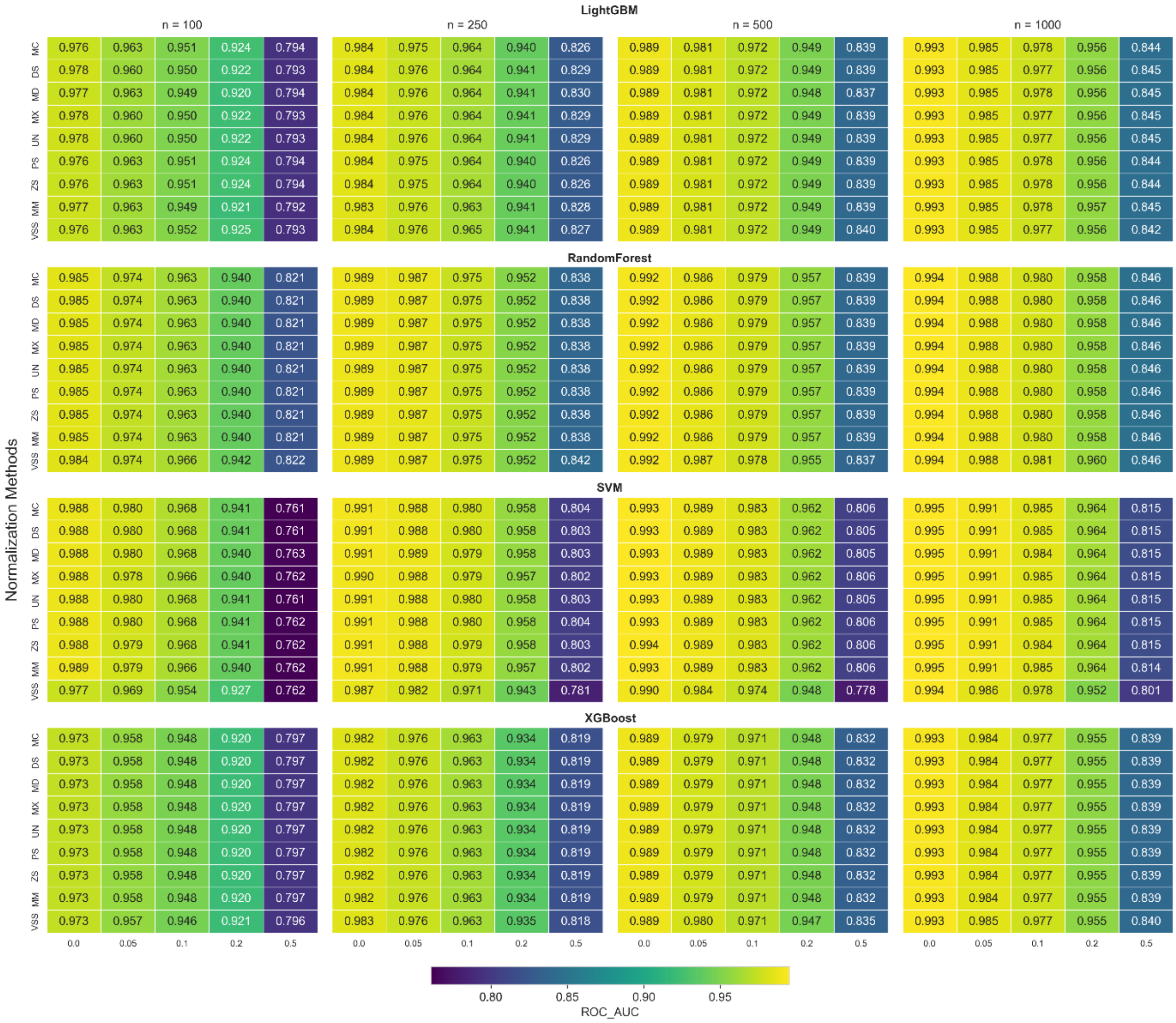


**Figure 1:** Heatmap of ROC-AUC scores across all simulation scenarios. The color gradient from purple (low AUC) to yellow (high AUC) visually represents the performance, clearly showing how performance degrades with increasing outlier ratios (x-axis) and varies across algorithms and normalization methods (y-axis).

As you can see from the heatmap, the right side of each panel, which represents the highest outlier ratios, is consistently darker, indicating poorer performance. But you can also see some methods and algorithms seem to resist this degradation better than others.

### 3.2. Performance of Normalization Methods

A detailed investigation underscores the critical importance of normalization strategy selection when processing heterogeneous or non-normally distributed data (Vafaei et al., 2022). Table 2 shows the average performance of each normalization method as the outlier ratio increases.

**Table 2.** Mean ROC-AUC of Normalization Methods by Outlier Ratio.

| Outlier Ratio | DS | MC | MD | MM | MX | PS | UN | VSS | ZS |
|---|---|---|---|---|---|---|---|---|---|
| 0% | 0.9756 | 0.9756 | 0.9755 | 0.9756 | 0.9756 | 0.9756 | 0.9756 | 0.9746 | 0.9756 |
| 5% | 0.9708 | 0.9708 | 0.9712 | 0.9707 | 0.9707 | 0.9708 | 0.9708 | 0.9691 | 0.9707 |
| 10% | 0.9609 | 0.9609 | 0.9621 | 0.9609 | 0.9609 | 0.9609 | 0.9609 | 0.9589 | 0.9609 |
| 20% | 0.9361 | 0.9361 | 0.9385 | 0.936 | 0.9361 | 0.936 | 0.9361 | 0.933 | 0.936 |
| 50% | 0.8133 | 0.8132 | 0.8194 | 0.8131 | 0.8133 | 0.8132 | 0.8133 | 0.8092 | 0.8132 |

On clean data, the choice of normalization method yielded negligible performance variance. But as soon as we started adding outliers, the MD method consistently outperformed other techniques (Tseng et al., 2025). This provides compelling evidence that the utilization of robust statistics renders the method more effective at mitigating the distorting effects of outliers (Tseng et al., 2025).

### 3.3. Algorithmic Resilience

We also saw some big differences between the algorithms themselves. Table 3 summarizes their performance.

**Table 3.** Performance Comparison of Algorithm Groups.

| Outlier Ratio | Tree-Based (Average) | SVM |
|---|---|---|
| 0% | 0.9743 | 0.979 |
| 5% | 0.968 | 0.9783 |
| 10% | 0.9579 | 0.9689 |
| 20% | 0.9333 | 0.9429 |
| 50% | 0.8201 | 0.7908 |

Interestingly, on clean data, the SVM was actually the best performer. But it was also the most fragile (Thakur et al., 2025). At the 50% outlier ratio, its performance had experienced a precipitous decline, while the tree-based models, though also impacted, held up much better (Wyatt et al., 2022). This crossover confirms that algorithms like Random Forest and XGBoost are just structurally more robust to outliers than SVM (Choudhury et al., 2024).

### 3.4. Critical Interaction

The most interesting result, though, is what happens when you combine a sensitive algorithm like SVM with a robust scaling method like MD. We found that MD scaling effectively preserves SVM stability. At 50% contamination, using MD scaling boosted SVM's performance significantly compared to using the standard Z-Score. This shows that a strategic preprocessing selection can mitigate the limitations of a powerful but sensitive algorithm from failing in a noisy environment (Ashraf et al., 2024).

## 4. Discussion

The findings underscore the impact of scaling methodologies on model robustness, especially in the presence of the non-Gaussian distributions typical of field-acquired data (Çetin and Yıldız, 2022). The efficacy of normalization depends not merely on its application, but on the specific methodology employed. The results underscore the value of robust statistics in the preprocessing pipeline (Bilal et al., 2022).

The empirical success of Median/MAD scaling within high-outlier environments aligns with theoretical expectations regarding robust estimators. The median and MAD are theoretically established robust estimators. What our study does is provide strong, broad empirical evidence for this, showing it holds true across different data sizes, feature counts, and for different ML algorithms (Tseng et al., 2025). The traditional methods like Z-Score and Min-Max exhibited significant instability when the data got noisy, which should be a critical cautionary indicator for anyone using them as a default (Alshdaifat et al., 2021).

It is acknowledged that the performance of kernel-based methods, particularly SVM, is heavily influenced by hyperparameter settings. For instance, adjusting the regularization parameter (C) can relax the margin to accommodate outliers (soft margin). However, the results demonstrate that even with default settings, the choice of normalization is a dominant factor in model stability. Using a robust scaler like Median/MAD provides a foundational layer of resilience, ensuring that the model performs adequately even before expensive hyperparameter optimization processes are applied. This suggests that robust normalization should be viewed as a prerequisite step that stabilizes the optimization landscape for subsequent tuning.

We also confirmed the well-known structural differences in how algorithms handle outliers. The tree-based models (XGBoost, Random Forest, LightGBM) exhibited inherent resilience (Suenaga et al., 2023). This makes sense, as their splitting decisions are less affected by extreme values than SVM's hyperplane, which is influenced by every point (Zhang and Gong, 2020). Li and Zhang (2023) proposed a high-dimensional ensemble outlier detection approach based on entropy-weighted subspaces, demonstrating that the robustness of ensemble methods importantly depends on data dimensionality (Li and Zhang, 2023). It seems that in high-dimensional, noisy data, the natural robustness of the tree models becomes even more of an advantage (Defilippis et al., 2024).

However, the most significant finding here is the synergy between preprocessing and algorithm choice. Algorithm and scaler selection cannot be performed in isolation; rather, they must be treated as interdependent components of the modelling pipeline. Our finding that MD scaling can substantially mitigate the otherwise pronounced fragility of SVM provides a clear example (Ashraf et al., 2024). It suggests that practitioners can still use powerful but sensitive algorithms like SVM, as long as they're smart about their preprocessing (Thakur et al., 2025). Ashraf et al. (2024) tried to tackle this by developing a hybrid rule-extraction system, but our results suggest that simply choosing a more robust scaler might be a more direct and effective solution (Ashraf et al., 2024). This is a practical finding.

It is worth noting that while ROC-AUC is the optimal metric for assessing general separability in balanced scenarios, specific real-world applications with high class imbalance or asymmetric misclassification costs might benefit from alternative metrics such as the F1-score or Precision-Recall curves. However, in this controlled simulation with balanced classes, ROC-AUC offers the most consistent benchmark for comparing normalization robustness.

Looking at the broader context, our work fits into a growing body of research that emphasizes the need for more rigorous and systematic evaluation of ML pipelines (Bischl et al., 2021). A 2021 paper by Bischl et al. on the OpenML platform argued for exactly this kind of large-scale, systematic comparison. Recent systematic reviews have emphasized the persistent challenges of data quality in real-world machine learning applications and the need for more robust preprocessing strategies (Sujon et al., 2024; Kalinina et al., 2025). Our study answers that call directly, providing a clear, data-driven comparison of scaling methods under precisely controlled conditions of data contamination (Glotsos et al., 2025). Ultimately, this study is a reminder that this study reinforces the 'No Free Lunch' theorem's relevance to preprocessing applies just as much to preprocessing as it does to modeling. There's no single best method for all situations. The optimal choice depends on the expected level of data contamination, the size and shape of the data, and the chosen algorithm (Li et al., 2023).

## 5. Conclusions

The systematic simulation study conclusively demonstrated that the choice of normalization method can have a huge impact on the performance of machine learning models, especially when the data is contaminated with outliers. We found that the Median/MAD scaling method is the optimal method in terms of robustness, providing a crucial buffer that can keep a model from failing (Tseng et al., 2025). The study also reinforces the idea that there’s no “one-size-fits-all” solution in machine learning.

Furthermore, results indicate that the optimal approach is inherently context-dependent, and an informed practitioner needs to think carefully about the interplay between their data, their preprocessing methods, and their choice of algorithm. The clear, empirical evidence provided in this study will help practitioners make more informed decisions and build more robust and reliable models. A limitation of this study is the use of fixed hyperparameters. Future research should investigate the interaction between robust normalization methods and automated hyperparameter optimization strategies (e.g., Bayesian optimization) to determine if tuning can further mitigate the effects of extreme outlier contamination observed in this study.

## 6. Conflict of Interest

The authors declare no conflict of interest.

## References


Alshdaifat E., Alshdaifat DA., Alsarhan A., Hussein F., El Salhi SMFS. The effect of preprocessing techniques, applied to numeric features, on classification algorithms' performance. Data 2021; 6(2): 11.

Ashraf MWA., Habib I., Akram MU., Al Ghamdi M. A hybrid approach using support vector machine rule extraction for anomaly detection. Sci Rep 2024; 14(1): 27058.

Bilal M., Ali G., Iqbal MW., Anwar M., Malik MSA., Kadir RA. Auto-prep: efficient and automated data preprocessing pipeline. IEEE Access 2022; 10: 107764-107784.

Bischl B., Casalicchio G., Feurer M., Gijsbers P., Hutter F., Lang M., Mantovani RG., van Rijn JN., Vanschoren J. OpenML benchmarking suites. arXiv [Preprint] 2021. arXiv:1708.03731. Available from: https://arxiv.org/abs/1708.03731

Çetin V., Yıldız O. A comprehensive review on data preprocessing techniques in data analysis. Pamukkale Univ J Eng Sci 2022; 28(2): 299-312.

Choudhury A., Mondal A., Sarkar S. Decision tree based machine learning algorithms: A comparative study of Random Forest, AdaBoost, XGBoost and LightGBM. Eur Phys J Spec Top 2024; 233(15): 2425-2463.

Dash CSK., Behera R., Dehuri S., Ghosh A. An outliers detection and elimination framework in classification task of data mining. Decis Anal J 2023; 6: 100164.

Defilippis L., Xu Y., Girardin J., Troiani E., Erba V., Zdeborová L., Loureiro B., Krzakala F. Scaling laws and spectra of shallow neural networks in the feature learning regime. arXiv [Preprint] 2025. arXiv:2509.24882.

Glotsos D., Kostopoulos S., Liaparinos P. Investigating student success rates in biomedical engineering education using machine learning and descriptive statistics. WSEAS Trans Adv Eng Educ 2025; 22: 107-113.

Huang L., Qin J., Zhou Y., Zhu F., Liu L., Shao L. Normalization techniques in training DNNs: methodology, analysis and application. IEEE Trans Pattern Anal Mach Intell 2023; 45(8): 10173-10196.

Kalinina I., Gozhyj A., Bidyuk P., Gozhyi V., Korobchynskyi M., Nadraga V. A systematic approach to data normalization and standardization in machine learning problems. In: Babichev S., Lytvynenko V., editors. Lecture Notes in Data Engineering, Computational Intelligence, and Decision-Making. Vol 2. Lecture Notes on Data Engineering and Communications Technologies; vol 244. Cham: Springer 2025; p. 206-219.

Kandanaarachchi S., Muñoz MA., Hyndman RJ., Smith-Miles K. On normalization and algorithm selection for unsupervised outlier detection. Data Min Knowl Discov 2020; 34: 309-354.

Khan H., Rasheed MT., Zhang S., Wang X., Liu H. Empirical study of outlier impact in classification context. Expert Syst Appl 2024; 256: 124953.

Kord A., Zand M., Zand S. Academic course planning recommendation using machine learning models. Educ Inf Technol 2024; 29: 1245-1268.

Kord A., Aboelfetouh A., Shohieb SM. Academic course planning recommendation and students' performance prediction multi-modal based on educational data mining techniques. J Comput High Educ 2025.

Lartey C., Liu J., Asamoah RK., Greet C., Zanin M., Skinner W. Effective outlier detection for ensuring data quality in flotation data modelling using machine learning (ML) algorithms. Minerals 2024; 14(9): 925.

Li X., Liu F. A mathematical comparison of data assimilation and machine learning in earth system state estimation from a Bayesian inference viewpoint. Information Geography 2025; 1(1): 100001.

Li Z., Zhang L. An ensemble outlier detection method based on information entropy-weighted subspaces for high-dimensional data. Entropy 2023; 25(8): 1185.

Mishra P., Singh U., Sahoo S. New data preprocessing trends based on ensemble of multiple preprocessing techniques. TrAC Trends Anal Chem 2020; 132: 116045.

Scikit-learn developers. Compare the effect of different scalers on data with outliers. In: scikit-learn: Machine Learning in Python. Version 1.8.0 2025. Available from: https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html

Singh D., Singh B. Feature wise normalization: An effective way of normalizing data. Pattern Recognit 2022; 122: 108307.

Suenaga D., Takase Y., Abe T., Orita G., Ando S. Prediction accuracy of Random Forest, XGBoost, LightGBM, and artificial neural network. Structures 2023; 50: 1252-1263.

Sujon KM., Hassan RB., Towshi ZT., Othman MA., Samad MA., Choi K. When to use standardization and normalization: Empirical evidence from machine learning models and XAI. IEEE Access 2024; 12: 135300-135314.

Thakur S., Tiwari VK., Agrawal J. Performance analysis of linear kernel support vector machine models on real-world datasets. Int J Adv Netw Appl 2025; 17(1): 6753-6760.

Tseng CY., Salguero JA., Breidenbach JD., et al. Evaluation of normalization strategies for mass spectrometry-based multi-omics datasets. Metabolomics 2025; 21: 98.

Vafaei N., Ribeiro RA., Camarinha-Matos LM. Comparison of normalization techniques on data sets with outliers. Int J Decis Support Syst Technol 2022; 14(1): 1-17.

Wyatt M., Radford B., Callow N., Brown L. Using ensemble methods to improve the robustness of deep learning for image classification. Methods Ecol Evol 2022; 13(6): 1317-1328.

Zhang D., Gong Y. The comparison of LightGBM and XGBoost coupling factor analysis and prediagnosis of acute liver failure. IEEE Access 2020; 8: 220990-221003.

Zhu X., Li Y. Robust feature scaling for time series forecasting using interquartile range. Expert Syst Appl 2023; 228: 120401.