FSDA: Tackling Tail-Event Analysis in Imbalanced Time Series Data with Feature Selection and Data Augmentation

Raphaël Krief RAPHAEL.KRIEF@ENS-PARIS-SACLAY.FR
Eric Benhamou ERIC.BENHAMOU@AIFORALPHA.COM
Beatrice Guez BEATRICE.GUEZ@AIFORALPHA.COM
Jean-Jacques Ohana JEAN-JACQUES.OHANA@AIFORALPHA.COM
David Saltiel DAVID.SALTIEL@AIFORALPHA.COM
Rida Laraki RIDA.LARAKI@LAMSADE.DAUPHINE.FR
Jamal Atif JAMAL.ATIF@LAMSADE.DAUPHINE.FR

Editors: Nuno Moniz, Paula Branco, Luis Torgo, Nathalie Japkowicz, Michał Wozniak and Shuo Wang.

Abstract

Efficient management of imbalanced time series data is of paramount importance when data located in the tails, particularly extreme values, have a substantial influence on predictive outcomes. This paper introduces FSDA (Feature Selection and Data Augmentation), a combined approach of feature selection and data augmentation, to address this issue. FSDA aims to identify the most predictive features for tail data, which may exhibit different sensitivities compared to the rest of the dataset. Data augmentation, a conventional technique for handling imbalanced data, is employed to enhance the accuracy of machine learning regression methods. Augmented information is strategically incorporated using time-warping and drift methods to maintain the temporal integrity of the data. Empirical evidence based on a use case in financial data reveals that FSDA consistently outperforms feature selection (FS) and data augmentation (DA) methods across all percentiles ranging from 85 to 99, demonstrating its efficacy in managing imbalanced time series data and improving predictive accuracy.

Keywords: Imbalanced time series, features selection, data augmentation

1. Introduction

1.1. Motivations

Data imbalance is a prevalent and inherent phenomenon in real-world contexts. Imbalance is characterized by data that does not adhere to an ideally uniform distribution among categories but rather showcases skewed distributions characterized by a long tail. This leads to notably fewer instances for specific target values Buda et al. (2018); Liu et al. (2019).

Such a phenomenon precipitates substantial challenges, particularly within disciplines where tail distributions have considerable ramifications. Deep recognition models serve as a prime example of this, being notably influenced by data imbalance. This, in turn, has prompted the creation of a myriad of techniques aimed at remedying this concern Huang et al. (2019); Cao et al. (2019); Liu et al. (2019); Cui et al. (2019); Tang et al. (2020).

Notwithstanding, current approaches to learning from imbalanced data predominantly concentrate on targets with categorical indices, wherein the targets correspond to distinctive classes. The term "imbalanced data" is frequently employed in relation to classification, less so with regression. It alludes to a situation where classes in a classification dataset are not evenly represented. An illustrative example would be a dataset comprised of 1000 samples, of which 950 are class A, leaving only 50 as class B. This presents a problem as machine learning models might exhibit bias towards the majority class, resulting in subpar classification performance on the minority class.

In the field of regression analysis, the term "imbalanced data" is not commonly used in a comparable manner. However, it can refer to situations where the distribution of the target variable (y) is skewed or uneven. For example, let's consider a dataset of housing prices where the majority of houses fall within the range of \$100,000 to \$200,000, while only a small number of houses are valued above \$1,000,000. In such cases, the dataset can be considered imbalanced, posing a greater challenge for the model to accurately predict prices for the more expensive houses. This challenge arises due to the limited number of data points available to train the model on these high-value properties.

Similarly, in the context of financial markets, we can also encounter imbalanced regression scenarios when predicting stock returns or developing a refined version of the Sharpe ratio for equity markets. In the case of stock return prediction, the imbalanced nature is evident in the distribution of returns, where the majority of stocks demonstrate relatively modest returns, while a few stocks exhibit significant positive or negative returns. This imbalance in the return distribution poses a difficulty for regression models in accurately predicting extreme returns. In both housing price prediction and stock return prediction, the presence of imbalanced data poses challenges for regression models to effectively capture and predict the outcomes that deviate significantly from the majority distribution.

Indeed, many real-world tasks necessitate dealing with continuous and potentially infinite target values. This is particularly conspicuous in fields such as finance, where objectives may include evading crash events or identifying highly profitable situations, or in computer vision, where the estimation of an individual's age based on visual appearances relies on a continuous target which can demonstrate notable imbalance. Analogous challenges surface in medical applications, where health metrics like heart rate, blood pressure, and oxygen saturation are continuous and often exhibit skewed distributions across patient populations.

In this research, we are interesting in not only playing with traditional data augmentation but also selecting appropriate variables or features through features selection to tackle the issue of imbalanced data for regression. We empirically observe that combining conventional data augmentation with features selection improves the treatment of imbalanced data.

1.2. Problem Definition

Let's denote our dataset as D, which contains n instances. Each instance can be represented as a pair (x_i, y_i), where x_i is the feature vector and y_i is the corresponding target value. For simplicity, let's consider a regression task where the target variable y is continuous. We can define the empirical distribution of the dataset as $p_{data}(y)$.

The imbalance in the dataset can be characterized by the skewness of this empirical distribution. Skewness, measured by the traditional recentered third moment: $\mathbb{E}\left[\left(\frac{y-\mu}{\sigma}\right)^3\right]$, where μ and σ are the mean and standard deviation of the target values y in D, and $\mathbb{E}$ denotes the expectation is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined.

The task of a regression model, in this case, can be defined as to learn a function f from the feature space to the target space, $f : x \to y$, such that it minimizes some loss function $L(y, f(x))$. For instance, in the case of linear regression, $L(y, f(x))$ could be Mean Squared Error (MSE), defined as:

$$L(y, f(x)) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

However, an important caveat in this modeling process arises when dealing with imbalanced data. In such scenarios, the model's performance may degrade, particularly on the tail of the distribution. This is an issue of significant concern within the domain of financial markets, where the tail-end of the distribution can dramatically influence the efficacy of the resultant strategy. Utilizing the traditional Mean Squared Error (MSE) in these circumstances may not adequately account for the tail distribution, a component of the data that carries critical weight in financial market analyses.

In some cases, the data is not only imbalanced but also scarce, especially for certain target values. Under such circumstances, we may need to not only be able to correctly learn the mapping from the feature space to the target space but also require some subsampling techniques to further increase the number of targets such as to improve the performance of our regression model.

Moreoer, feature selection may play an important role when dealing with imbalanced data in regression, a topic that often receives inadequate attention in traditional machine learning literature.

The features used for regression analysis significantly impact the model's ability to accurately capture relationships, especially when data is imbalanced. In such cases, irrelevant or redundant features can obscure the information provided by under-represented target values, leading to sub-optimal model performance.

To incorporate feature selection into our framework, let's denote the original feature space as $\mathcal{X}$, and a subset of selected features as $\mathcal{X}_s \subseteq \mathcal{X}$.

The task of feature selection can be viewed as finding the optimal feature subset $\mathcal{X}_s$ that minimizes the loss function when using these features for training our regression model. We can denote this as follows:

$$\mathcal{X}_s = \arg\min_{\mathcal{X}' \subseteq \mathcal{X}} \frac{1}{|D_s|}\sum_{i \in D_s} w_i(y_i - f(x_i, \mathcal{X}'))^2$$

where $f(x_i, \mathcal{X}')$ denotes the model's output when trained on the features $\mathcal{X}'$.

The challenge will therefore be to incorporate these two techniques: data augmentation and features selection to be able to improve the accuracy of our regression exercise on large values.

1.3. Related Works

Traditionally, the imbalance problem has been predominantly addressed in the context of classification tasks, with limited focus on imbalanced regression, which is less explored in the literature. However, it is equally important to address imbalanced data in regression scenarios, as the presence of imbalanced data can significantly impact the performance and generalization capabilities of regression models.

Several works have started to delve into the challenges of imbalanced regression. For instance, Yang et al. (2021) propose an approach called "Delving into Deep Imbalanced Regression," which focuses on effectively handling imbalanced regression through a deep learning framework. They propose novel loss functions and training strategies to improve the performance of regression models in the presence of imbalanced data.

In addition to deep learning approaches, non-linear gradient boosting methods have also been explored for imbalanced regression. Frery et al. (2018) present a technique called "Non-Linear Gradient Boosting for Class-Imbalance Learning," which adapts the boosting algorithm to effectively handle imbalanced regression problems. Their approach combines a gradient boosting framework with non-linear transformations to better capture the underlying patterns in imbalanced regression data.

Furthermore, addressing imbalanced regression requires specific methods that can effectively tackle the problem of imbalance in the data. Branco et al. (2017) propose "SMOGN," a pre-processing approach specifically designed for imbalanced regression tasks. Instead of incorporating feature selection, SMOGN blends over-sampling of rare cases with under-sampling of common cases, focusing on the distribution of the target variable rather than the features. This approach improves the performance of regression models on imbalanced datasets by providing a better representation of the less frequent, but often more important, cases.

Overall, while the research community has primarily focused on managing imbalanced data in classification tasks through techniques like under-sampling and over-sampling, it is crucial to extend these techniques to imbalanced regression scenarios. Additionally, incorporating feature selection methods alongside imbalance handling techniques can further enhance the performance and interpretability of regression models on imbalanced datasets.

2. Contribution

To effectively mitigate the challenge posed by an imbalanced time series, we propose a combined approach that incorporates both feature selection and data augmentation. By strategically varying the order in which these techniques are applied, we can discern the predominant factor contributing to improved performance. Let us present these two techniques in details.

2.1. Data Augmentation

Data augmentation is a technique used to artificially increase the size of the imbalanced time series dataset. The main idea is for each minority class sample to generate synthetic samples by applying interpolation to similar samples and extrapolation to nearby samples. This approach helps in creating a more balanced dataset by increasing the number of samples in the minority class without introducing bias. The algorithm for data augmentation is presented in Algorithm 1. Let us present two traditional data augmentations: TimeWarp and Drift.

Algorithm 1: Data Augmentation Algorithm

```
1. Input: Imbalanced time series dataset $D$
2. Output: Augmented dataset $D'$
3. Initialize empty dataset $D'$
4. For each sample $x$ in $D$, do:
    (a) If $x$ belongs to minority class, then:
        i. Generate $n$ synthetic samples using interpolation and extrapolation
        ii. Add synthetic samples to $D'$
    (b) Else:
        i. Add $x$ to $D'$
5. Return $D'$
```

2.2. TimeWarp

TimeWarp is a non-linear transformation of the time axis. It warps the time series by rescaling the time axis with a randomly generated warping function. The transformation can be formally described as follows:

Let $X(t)$ be a given time series. The TimeWarp method transforms $X(t)$ to $X'(t')$ where $t' = f(t)$. Here, $f(t)$ is a continuous, strictly increasing function that defines the warping of the time axis. An example of such a function could be $f(t) = at$, where a is a random variable. The exact form of $f(t)$ may vary depending on the specifics of the implementation, and is usually designed to create realistic warping of time series data.

2.3. Drift

Drift is an another augmentation technique that adds a trend to the time series data. It generates a new time series by adding a linear or non-linear trend to the original time series. This method could be formulated as follows:

Let $X(t)$ be a given time series. The Drift method transforms $X(t)$ to $X'(t) = X(t) + d(t)$, where $d(t)$ is a drift term. In our case, we use a simple linear function as a drift term such as $d(t) = bt$, where b is a random variable sampled from a uniform distribution between 0.1 and 0.5.

3. Feature Selection Methods

In this section, we discuss two popular methods for feature selection: Recursive Feature Elimination (RFE), and Lasso. These techniques aim to identify the most relevant features for a given predictive task, effectively reducing dimensionality and improving model performance.

3.1. Recursive Feature Elimination (RFE)

RFE is a feature selection method that recursively eliminates less important features from the dataset. Usually done backward, it can also be done forward. RFE begins by training a model on the full feature set. Features are then ranked by their importance scores and the feature with the lowest score is removed. This iterative process continues until the remaining features reach a predetermined number.

In mathematical terms, if the input feature matrix is denoted by $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n samples and p features, the objective of RFE is to select a subset of features $\mathbf{X}_{\text{selected}} \in \mathbb{R}^{n \times k}$, where $k < p$. The selected subset of features should maximize the model's performance metric, such as accuracy or mean squared error.

The RFE algorithm can be summarized by the Algorithm 2.

Algorithm 2: Recursive Feature Elimination (RFE)

```
1. Input: Feature matrix $\mathbf{X}$, Target variable $\mathbf{y}$, Number of features $k$
2. Output: Selected feature matrix $\mathbf{X}_{\text{selected}}$
3. Initialize $\mathbf{X}_{\text{selected}} \leftarrow \mathbf{X}$
4. While num_features($\mathbf{X}_{\text{selected}}$) $> k$:
    (a) Train a model on $\mathbf{X}_{\text{selected}}$ and $\mathbf{y}$
    (b) Calculate feature importance scores
    (c) Remove the feature with the lowest score from $\mathbf{X}_{\text{selected}}$
5. Return $\mathbf{X}_{\text{selected}}$
```

RFE, by fitting a model and successively removing the weakest feature(s), reduces the feature set to a specified limit. The features are ranked based on the model's feature importances attributes, and the method recursively eliminates features to eliminate collinearity and dependencies within the model.

3.2. Lasso

Least Absolute Shrinkage and Selection Operator (LASSO) implements both variable selection and regularization, thereby enhancing the accuracy and interpretability of the model it generates. A penalty term is incorporated into the least squares objective, thereby effectively reducing less important features' coefficients to zero. LASSO's mathematical representation is given by:

$$\hat{\beta}^{lasso} = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{1}$$

where $\hat{\beta}^{lasso}$ denotes the estimated parameters, y_i represents the response variables, β_0 and β_j represent the model's parameters, x_{ij} signifies the predictor variables, and λ controls the amount of regularization.

In the case of correlated variables, Lasso tends to select one variable from a group and disregards the rest. The steps involved in Lasso feature selection are as follows:

1. Train a Lasso regression model on the full feature set.
2. Obtain the magnitude of the feature coefficients.
3. Set a threshold and select the features with coefficients above the threshold.

3.3. Comparison of RFE and Lasso

To compare the effectiveness of RFE and Lasso, we can analyze three theoretical situations:

1. Sparsity: Lasso has built-in sparsity-inducing properties, meaning it tends to produce models with a small number of nonzero coefficients. This property makes Lasso particularly suitable for feature selection when the number of relevant features is expected to be small.
2. Consistency: Under certain assumptions, Lasso has been shown to consistently select the true relevant features as the sample size increases, even in high-dimensional settings. This property provides theoretical support for the reliability of Lasso's feature selection capabilities.
3. Collinearity Handling: Lasso performs well in the presence of multicollinearity, as it encourages shrinkage of correlated features towards zero. In contrast, RFE may struggle to select the most important features in highly collinear datasets.

Based on these theoretical arguments, Lasso is often preferred over RFE when dealing with high-dimensional datasets, especially when features are expected to be sparse or exhibit collinearity.

3.4. Problem with RFE in Presence of Multicollinearity

The problem with RFE in the case of multicollinearity arises due to the fact that it does not handle redundancy in features. If two features are highly correlated, RFE might keep both of them even though one could be discarded without loss of information.

Now, let's consider a hypothetical scenario where we have a dataset with p features that are highly correlated. If we denote the correlation as ρ, the collinearity can be expressed as:

$$\rho = \frac{\text{Cov}(X_i, X_j)}{\sigma_i \sigma_j} \tag{2}$$

where $\mathrm{Cov}(X_i,\ X_j)$ is the covariance between feature i and j, and σ_i and σ_j are the standard deviations of feature i and j respectively.

In case of perfect collinearity, ρ approaches 1, and as a result, the covariance matrix of the features $(X'X)$ approaches singularity. In such a case, RFE may fail to distinguish between the importance of features, as the features would appear equally important due to high correlation.

On the contrary, the L1 penalty in Lasso leads to sparsity, which means that it forces less important features' coefficients to zero, thereby effectively reducing the feature set. This happens regardless of the correlation between the features, making Lasso more effective in scenarios with high multicollinearity.

Hence, given the properties of Lasso, it is quite intuitive that LAsso tends to outperform RFE in situations with high multicollinearity.

4. Experiment

4.1. Experiment Objectives and Data Description

The objective of our study is to predict the Sharpe ratio, calculated over 30 periods as initially proposed in Sharpe (1975), for various assets. The Sharpe ratio (SR) is mathematically defined as the ratio of the excess return (R_{excess}) over the investment's volatility (σ): $SR = \frac{R_{\text{excess}}}{\sigma}$. In this formulation, R_{excess} stands for the discrepancy between the average return of the investment and the risk-free rate, while σ is indicative of the standard deviation of the investment's returns. The Sharpe ratio is praised for its simplicity and versatility, and has been extended to better account for drawdowns (Challet, 2017) and target diversified portfolios that perform well out of sample (Lopez de Prado, 2016). Because of its tractability, the Sharpe ratio is also commonly used to evaluate hedge funds and mutual funds Sharpe (1975) and Sharpe (1992). Moreover, one can compute the statistics of having a specific Sharpe ratio at a given time horizon, which enables inferences about whether the asset manager has real skill or is simply lucky with their reported Sharpe ratio (Benhamou et al., 2019a). Last but not least one can prove that maximizing the Sharpe ratio is equivalent to maximizing the Omega ratio for elliptic distributions (Benhamou et al., 2019b). In our experiments, we used 11 assets. The exhaustive inventory of the 11 financial assets evaluated in this study can be found in table 1.

The features utilised in our analysis are derived from various sources. Some originate from the assets themselves and are subject to transformations as detailed in table Table 2. Others are contextual variables, outlined in table Table 3, and undergo different feature transformations as elaborated in table Table 4, where we distinguish variables according to their sign.

Consequently, the dataset integrates daily financial metrics from April 2nd, 2008, to April 14th, 2023, amounting to a total of 38,456 data entries. By applying a traditional time series train split, the dataset bifurcates into a training set and a test set. The training set, active from April 2nd, 2008, to December 31st, 2018, encompasses 27,709 rows, while the test set, spanning from January 1st, 2019, to April 14th, 2023, includes 10,746 rows. Consequently, the training set comprises 72 percent of the entire dataset.

The data are primarily processed through two techniques: scaling and one-hot encoding. Scaling is crucial to standardize the range of input features, enabling the model to converge

Table 1: Asset Information

Number	Asset	Ticker	Description
1	S&P 500	SGBVRES1 Index	US Equities
2	Eurostoxx 50	SGBVRVG1 Index	EU Equities
3	Nikkei 250	SGBVRNK1 Index	Japan Equities
4	FTSE 100	SGBVRZ1 Index	UK Equities
5	10 year US Tnote	SGIXBTY Index	US 10yr Bond
6	10Yr Bund	SGIXBRX Index	EU 10yr Bond
7	10Year Gild	SGIXBGB Index	UK 10yr Bond
8	10Year Japanese Government Bond	SGIXBJB Index	Japan 10yr Bond
9	Brent	SGICCOSR Index	Brent
10	Gold	SGICGCSR Index	Gold
11	Copper	SGICHGSR Index	Copper

faster. One-hot encoding was employed to transform categorical data into a format that could be provided to the machine learning algorithm for more effective processing.

Furthermore, it is observed that the data are imbalanced, signifying a disproportionate ratio of observations in each class. This is not a surprise as tail events are quite common in finance (Sornette, 2003) or (Benhamou et al., 2021). Such a characteristic may result in a biased model that may not effectively generalize. Therefore, techniques such as data augmentation or feature selection are introduced to improve the model's performance in different cases.

The focus of our experiment lies in examining the mix of feature selection and data augmentation. We aim to determine whether each technique on its own, or their combined application, offers superior results for different portions of our upper tail data. This upper tail data is defined as values exceeding the q-th quantile, with q ranging from 0.85 to 0.99.

Table 2: Asset Features Transformations and Parameters

Data	**Transformation**	**Type**	**Parameters**
Asset	pct change	strictly positive	5, 10, 20, 60, 120, 250
Asset	std deviation	strictly positive	60, 125
Asset	sharpe	strictly positive	120, 250
Asset	distance to MA	strictly positive	250, 500
Asset	technical analysis	strictly positive	rsi 14, 30, stochRSI 14, 20, macd diff, signal

The data augmentation method involves applying transformations to generate synthetic data in order to increase the diversity and size of the dataset. In this specific case, the augmentation technique includes two transformations: TimeWarp and Drift.

Table 3: Common Features Transformations and Parameters

Data	**Type**	**Frequency**	**Category**
US Rates 10yr vs 2 yr	float	daily	Rates environment
EUR Rates 10yr vs 2 yr	float	daily	Rates environment
US 10yr Rate	float	daily	Rates environment
US Credit (CDS HY NA 5 year)	strictly positive	daily	Credit environment
EUR Credit (CDS HY NA 5 year)	strictly positive	daily	Credit environment
Correl Equity Bonds 20d	float	daily	Asset interactions
Correl Equity Bonds 60d	float	daily	Asset interactions
Correl Equity Bonds 120d	float	daily	Asset interactions
Correl Equity Bonds 250d	float	daily	Asset interactions
VIX Index	strictly positive	daily	Market stress
Volatility of VIX	strictly positive	daily	Market stress
Dollar Index	strictly positive	daily	Currencies
GDP Forecast (FED survey)	strictly positive	quarterly	Economist views
CPI Forecast (FED survey)	strictly positive	quarterly	Economist views

The TimeWarp transformation modifies the temporal structure of the data by warping it, introducing variations in the time series. This transformation allows for the generation of new instances with different temporal patterns, enhancing the variability of the dataset.

Table 4: Transformations and Parameters for common features

Type	**Transformation**	**Parameters**
strictly positive	percentage change	5, 10, 20, 60, 120, 250
strictly positive	std	60, 125
strictly positive	distance to standard Moving average	250, 500
float	difference	5, 10, 20, 60, 120, 250
float	standard deviations	60, 125
float	distance to standard Moving average	250, 500

Additionally, the Drift transformation introduces a drift effect to the data by modifying its baseline. The drift parameter is a random amount between 0.1 and 0.5. These two values are taken to be consistent with the order of magnitude of Sharpe ratios. The magnitude of the drift is randomly determined within a specified range (maximum drift), which controls the extent of the shift in the data. By incorporating this drift effect, the method generates additional instances with varying baseline levels, contributing to the augmentation of the dataset.

By applying these transformations, the data augmentation method aims to create synthetic data that captures different temporal patterns and baseline variations.

4.2. Results

Figure 1 reveals the imbalance exhibited by the exceptionally positive values of the Sharpe ratio. The distribution of the Sharpe ratio for the given dataset is delineated, highlighting segments of negative Sharpe ratios and those surpassing the q-th quantile, where q extends from 85 to 99. A notable observation from the figure is the 'fat tail' characteristic manifested by the values beyond the 99th quantile.

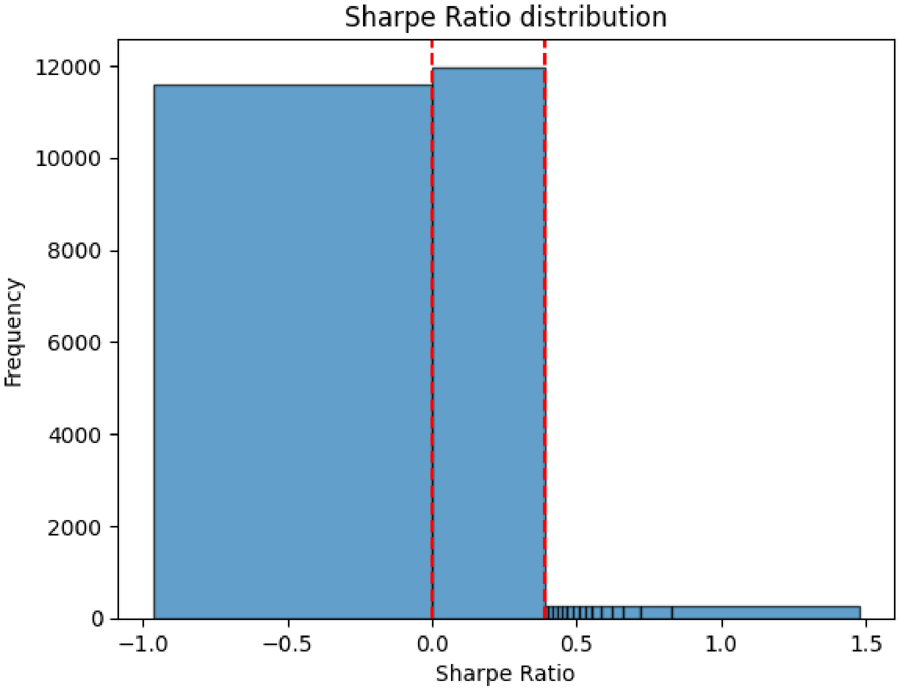

Figure 1: Distribution of Sharpe Ratios in Three Categories

Table 5 present the performance results of different methods, including Baseline, FS, DA, and FSDA, across different quantiles ranging from 85 to 99 percent.

The Baseline model serves as our baseline where a standard RFE method is done on an OLS model with 20 features but without any feature selection or data augmentation. It should logically underperform.

The second model entitled the Feature Selection (FS) model is the Lasso regression.

The third model entitled the Data Augmentation (DA) model, focuses solely on augmenting the available data

Last but not lest, the FS DA model, where both feature selection and data augmentation are combined is presented.

It is interesting to note that the FSDA method consistently achieves the lowest values among all methods, indicating its ability to produce more accurate results, across all quantiles.

Table 5: Model Comparison with Feature Selection and Data Augmentation

Quantile	**Baseline**	**FS**	**DA**	**FSDA**	**Best Method**
85	0.381	0.369	0.36	0.358	FSDA
86	0.39	0.378	0.368	0.366	FSDA
87	0.4	0.388	0.378	0.376	FSDA
88	0.411	0.399	0.389	0.387	FSDA
89	0.421	0.41	0.399	0.397	FSDA
90	0.43	0.418	0.407	0.405	FSDA
91	0.443	0.431	0.419	0.416	FSDA
92	0.456	0.443	0.431	0.428	FSDA
93	0.47	0.456	0.444	0.44	FSDA
94	0.489	0.476	0.465	0.461	FSDA
95	0.511	0.497	0.484	0.481	FSDA
96	0.541	0.526	0.512	0.509	FSDA
97	0.573	0.555	0.542	0.539	FSDA
98	0.606	0.587	0.573	0.569	FSDA
99	0.683	0.667	0.652	0.648	FSDA

The evaluation criterion used in this research is the Root Mean Square Error (RMSE), which is a widely adopted metric in regression tasks. RMSE measures the average magnitude of the differences between predicted values $\hat{y_i}$ and actual values y_i, providing an intuitive understanding of the model's predictive performance. It is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}$$

RMSE is particularly useful as it penalizes larger prediction errors more heavily, making it sensitive to outliers and deviations from the actual values.

Having established that the chosen loss function is RMSE, the consistent superiority of the FSDA method in achieving the lowest RMSE values reinforces its remarkable capability to generate more accurate predictions. This suggests that FSDA can be a valuable alternative to traditional data augmentation techniques, showcasing its potential to provide better predictions and enhance the overall model's performance.

4.3. Intuition and future work

The text highlights the benefits of combining data augmentation and feature selection in machine learning tasks:

1. **Data Augmentation:** Data augmentation techniques, such as Time Warping, can enhance machine learning model performance by amplifying and diversifying training data. This is particularly helpful with imbalanced datasets or extreme value instances.

However, this alone might not be sufficient when the dataset is noisy or contains irrelevant information, necessitating feature selection.

2. **Feature Selection:** Feature selection improves learning accuracy, reduces model complexity, and mitigates the 'curse of dimensionality' by eliminating less critical or irrelevant features. This is especially important when dealing with high-dimensional datasets.

Combining both techniques results in a machine-learning model that benefits from a balanced and representative dataset (via data augmentation) and an enhanced learning and generalization ability (through feature selection). While this combined approach is beneficial, its efficiency largely depends on the specific dataset and problem. Moreover, it might not fully resolve all challenges associated with imbalanced data or extreme values.

The combined approach could potentially bring significant advancements in the field of data augmentation techniques. However, it is important to substantiate this intuitive understanding with mathematical evidence. Future research should focus on providing mathematical proofs and empirical evaluations to validate the effectiveness of this combined approach in various machine-learning tasks.

5. Conclusion

In conclusion, the combined approach of feature selection and data augmentation presented in this study, called FSDA, demonstrates its effectiveness in managing imbalanced time series data and improving predictive accuracy. By identifying the most predictive features for tail data and strategically incorporating augmented information, FSDA outperforms traditional feature selection and data augmentation methods across various percentiles. This highlights its potential for practical applications, particularly in domains where extreme values in the tails play a crucial role in predictive outcomes.

For future work, further investigation could focus on exploring different combinations of feature selection and data augmentation techniques to enhance the performance of FSDA. Additionally, evaluating the generalizability of FSDA across diverse datasets and problem domains would provide valuable insights into its robustness and applicability. Such efforts will contribute to advancing the field of imbalanced time series analysis and facilitate more accurate predictions in real-world scenarios.

References

E. Benhamou, , D. Saltiel, Guez R., and Paris N. Testing sharpe ratio: luck or skill? *ArXiv*, 2019a.

E. Benhamou, B. Guez, and N. Paris. Omega and sharpe ratio. *ArXiv*, 2019b.

E. Benhamou, D. Saltiel, JJ. Ohana, and J. Atif. Detecting and adapting to crisis pattern with context based deep reinforcement learning. In *ICPR 2021 proceedings*, 2021.

P. Branco, L. Torgo, and R. P. Ribeiro. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017.

M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, pages 249–259, 2018.

K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.

D. Challet. Sharper asset ranking with total drawdown duration. *Applied Mathematical Finance*, pages 1–22, 2017.

Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

J. Frery, A. Habrard, M. Sebban, and L. He-Guelton. Non-linear gradient boosting for class-imbalance learning. In *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, volume 94, pages 38–51. PMLR, 2018.

C. Huang, Y. Li, C. L. Chen, and X. Tang. Deep imbalanced learning for face recognition and attribute prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. doi: 10.1109/TPAMI.2019.2914749.

Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

M. Lopez de Prado. Building diversified portfolios that outperform out of sample:. *The Journal of Portfolio Management*, 42:59–69, 07 2016. doi: 10.3905/jpm.2016.42.4.059.

W.F. Sharpe. Adjusting for risk in portfolio performance measurement. *Journal of Portfolio Management*, pages 29–34, Winter 1975.

W.F. Sharpe. Asset allocation: Management style and performance measurement. *Journal of Portfolio Management*, pages 7–19, Winter 1992.

D. Sornette. *Why stock markets crash.* Princeton University Press, 2003. doi: 10.1515/9781400839889.

K. Tang, J. Huang, and H. Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

Y. Yang, K. Zha, YC. Chen, H. Wang, and D. Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning (ICML)*, 2021.