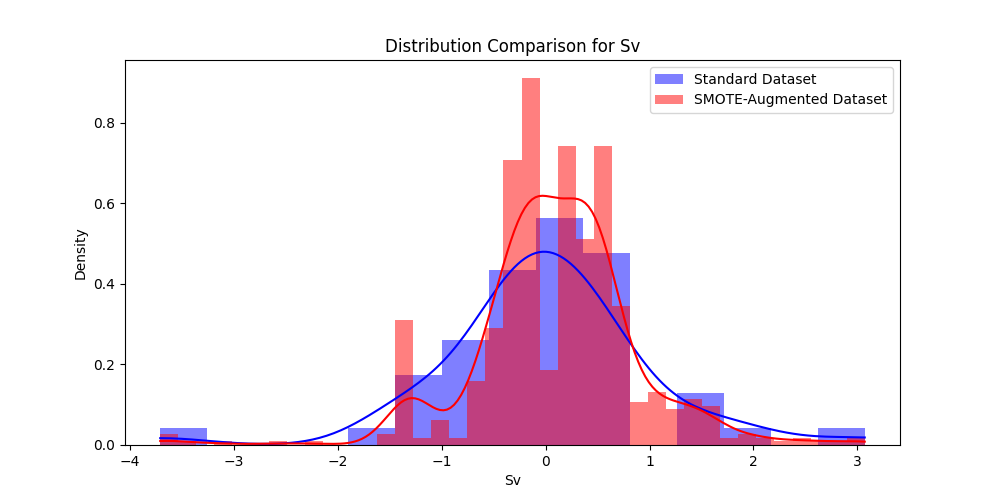
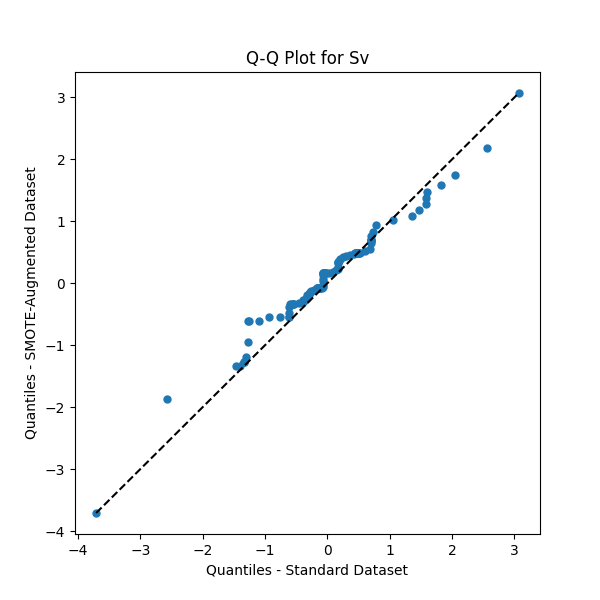
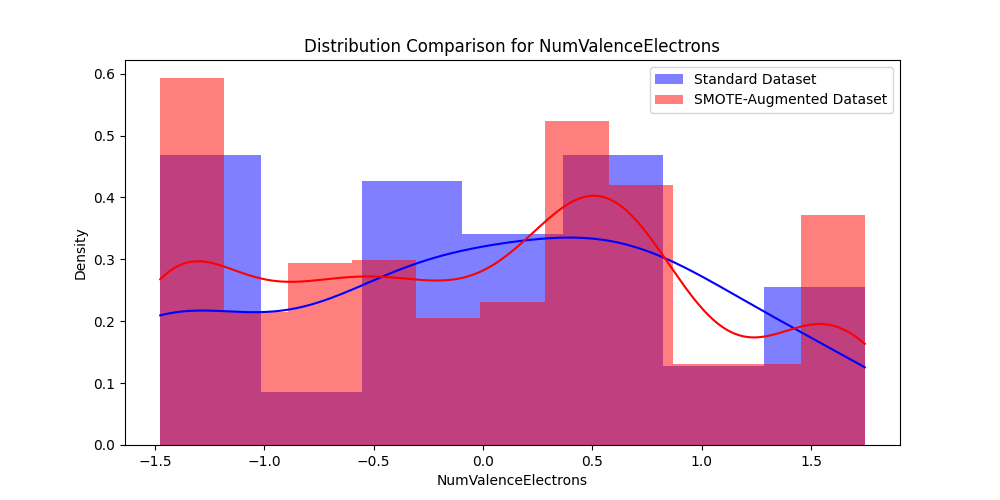
Histograms and Q-Q plots are essential tools for visualizing and comparing the distributional characteristics of datasets. Histograms provide a clear visual representation of data distribution by showing the frequency of data points within defined intervals, which helps in understanding the shape, spread, and central tendencies of data. They are particularly useful for assessing the effects of data augmentation techniques like SMOTE and Gaussian Copula, highlighting any significant changes such as shifts in central tendency or variability. On the other hand, Q-Q plots offer a nuanced comparison of two probability distributions by plotting their quantiles against each other, with points lying close to the line y = x indicating similarity between distributions. This makes Q-Q plots invaluable for examining finer details of distributional congruence, such as alterations in tail behavior or skewness introduced by the augmentation process. Together, these visual tools play a critical role in evaluating whether synthetic data augmentation maintains or distorts the statistical integrity of the original dataset, facilitating informed decisions regarding subsequent data processing steps. In the following distribution analysis, three out of the eight descriptors were examined. This selection was random, ensuring that the findings for these three descriptors are representative of the entire set. Consequently, the results are conclusively applicable to the remaining descriptors, accurately reflecting the overall distribution characteristics.

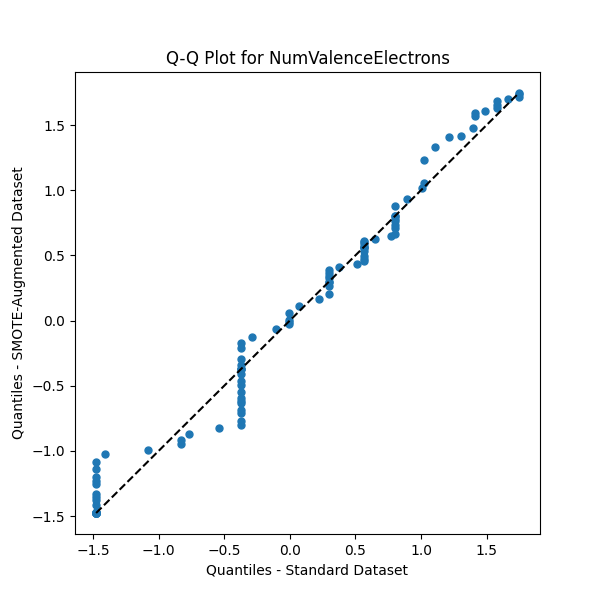
**a)**

**b)** 

***Fig.1 Comprehensive analysis of Sv distributions:*** *a) The first panel presents kernel density estimates comparing the original pre-SMOTE data with the SMOTE-augmented synthetic data for the feature Sv, highlighting subtle differences in density, particularly at the distribution tails; b) The second panel, a quantile-quantile plot, validates these observations, showing that the majority of data points closely adhere to the reference line (pre-SMOTE dataset), confirming substantial alignment in distributional characteristics, though with minor discrepancies at the extremes*

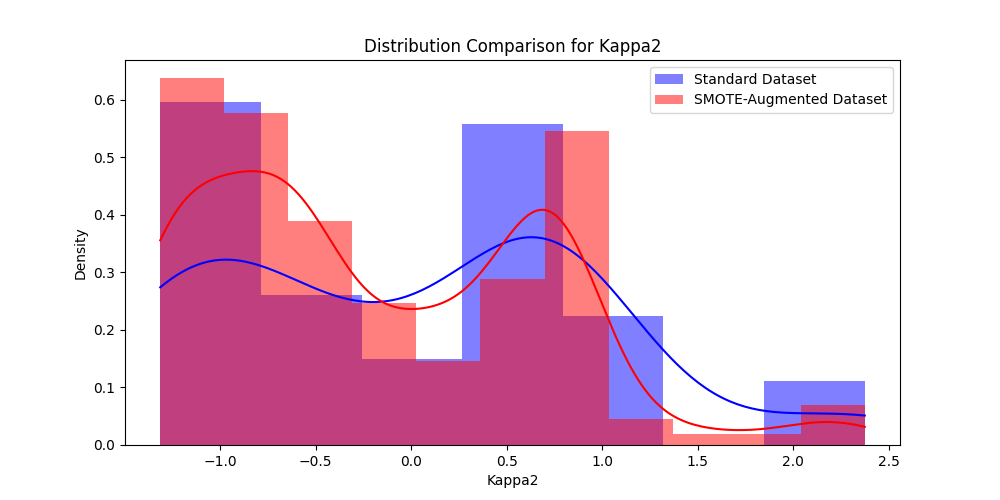
**a)**

**b)**

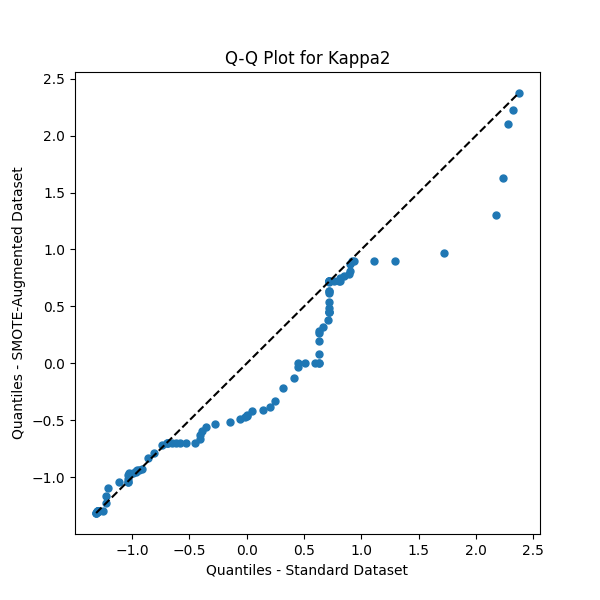


**Fig.2 Comprehensive analysis of NumValenceElectrons distributions:** a) The first panel presents kernel density estimates contrasting the original pre-SMOTE data with the SMOTE-augmented synthetic data for the feature NumValenceElectrons, highlighting subtle variations in density, particularly as it diverges slightly at the higher density areas and tails; b) The second panel, a quantile-quantile plot, corroborates these observations, with the majority of data points displaying close adherence to the reference line, signaling a good distributional match, albeit with slight discrepancies at the extremes

**a)**

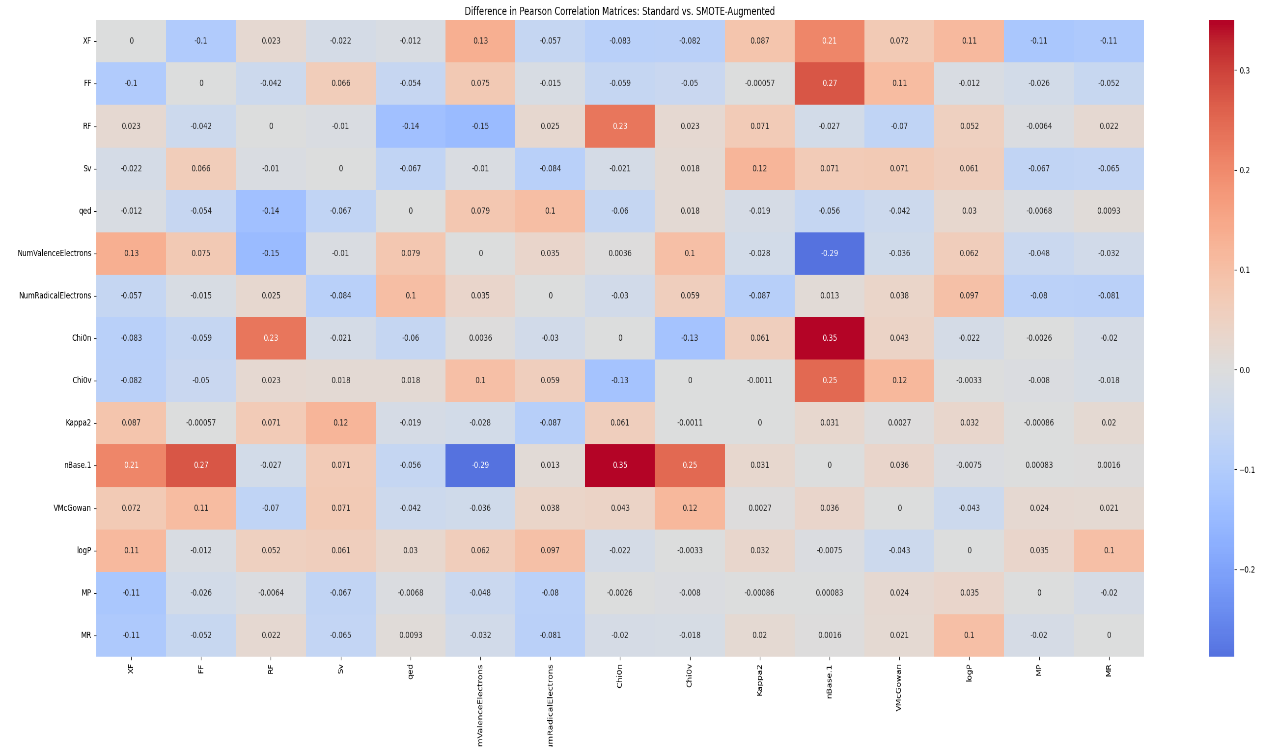


**b)**

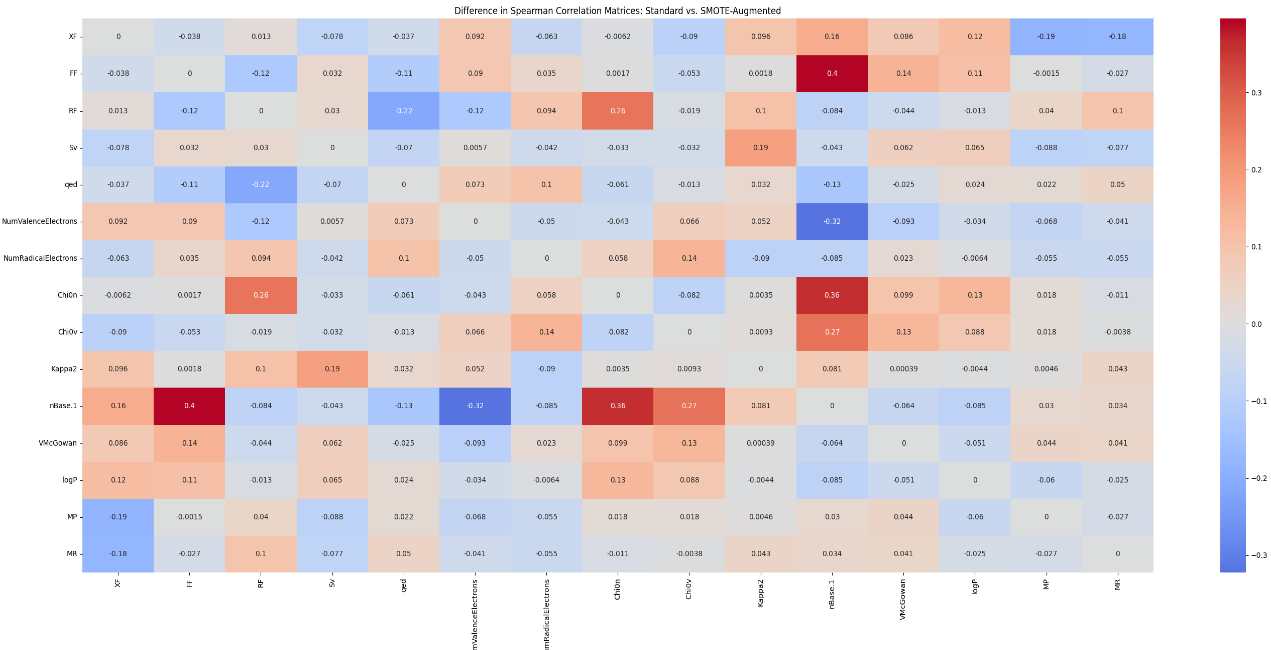


**Fig.3 Comprehensive analysis of Kappa2 distributions:** a) The first panel shows kernel density estimates for the feature Kappa2, comparing original pre-SMOTE data with SMOTE-enhanced synthetic data. While minor discrepancies are observed in higher density areas and tails, they do not substantially alter the overall distributional agreement; b) The second panel's quantile-quantile plot further supports these findings, with most data points aligning closely with the reference line, demonstrating a generally accurate replication of distributional characteristics, yet with slight deviations especially at the upper quantiles

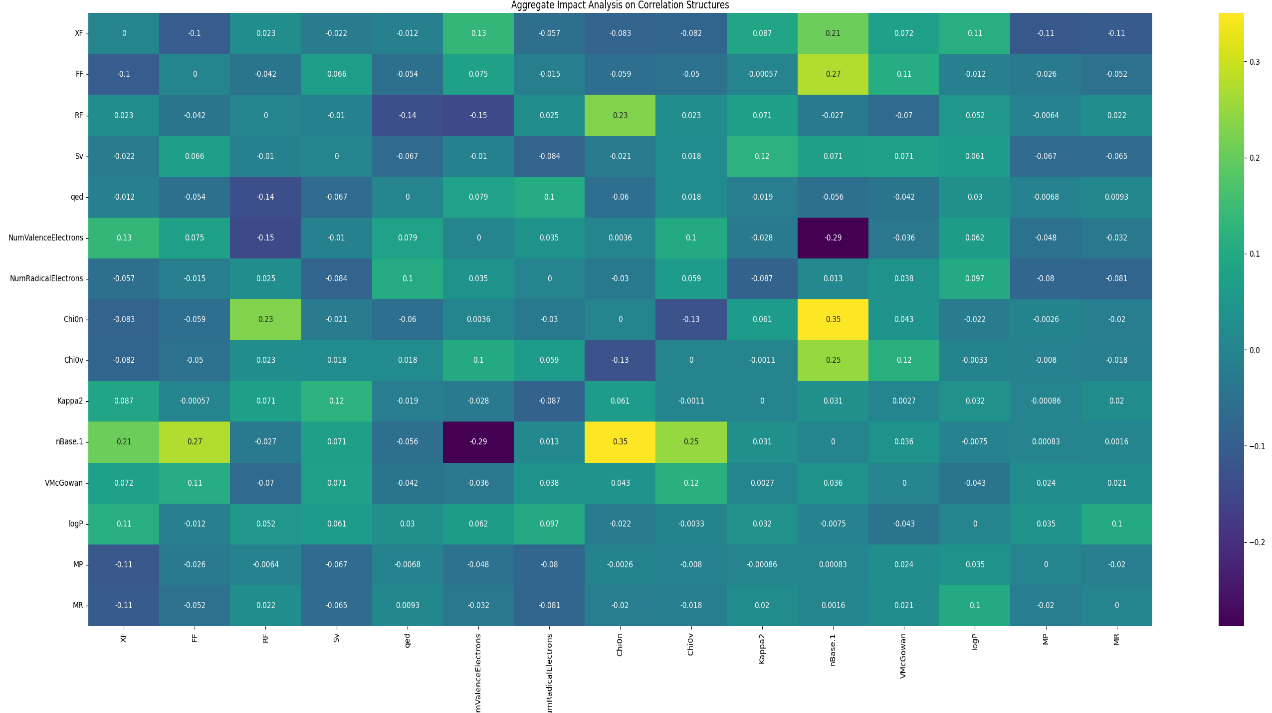
Pearson and Spearman correlation analyses are employed to assess linear and monotonic relationships, respectively, between descriptors in datasets before and after SMOTE augmentation. Pearson correlation quantifies linear associations, with coefficients indicating the strength and direction of these relationships; deviations in these correlations post-augmentation are visualized to highlight changes that could affect model dependencies. Conversely, Spearman correlation, which is insensitive to data distribution and more robust against outliers, evaluates rank-based relationships, allowing for the detection of alterations in non-linear and ordinal interactions crucial for analyses where linear assumptions fail. Additionally, a third comprehensive analysis of the differences in Pearson correlation matrices is conducted between the original and augmented datasets after all features have been processed, offering a holistic view of how linear relationships are affected across the entire dataset. This analysis uniquely aggregates and visualizes the cumulative alterations in Pearson correlation coefficients across the entire dataset post-augmentation, providing a comprehensive and detailed understanding of the global changes in linear relationships induced by the synthetic data augmentation technique, distinct from the feature-specific insights offered by individual Pearson or Spearman analyses.



**Fig.4 Differential analysis of Pearson Correlation Coefficients: Standard vs. SMOTE-Augmented data.** This heatmap visualizes the nuanced deviations in Pearson correlation coefficients between the original dataset and its SMOTE-based synthetic counterpart. Each cell in the heatmap quantifies the discrepancy in correlation between pairs of features, with values ranging from -0.3 to 0.35. The color gradient, shifting from cool blues (indicating a decrease in correlation relative to the standard dataset) to warm reds (indicating an increase), illustrates the extent of these variations. The majority of the coefficients exhibit minimal variations, especially for all the eight final features, highlighting the SMOTE method’s efficacy in closely replicating the original correlation structure



**Fig.5 Differential analysis of Spearman Correlation Coefficients: Original vs. SMOTE-Augmented data.** This heatmap provides a detailed comparison of the non-linear relationships between feature pairs, quantifying the impact of SMOTE augmentation on the original dataset's Spearman rank-order correlation coefficients. Each cell quantifies the change in correlation for a pair of features, with positive values colored in red indicating an increase in correlation, and blue values denoting a decrease, relative to the standard dataset. The range of changes, from -0.3 to 0.36, illustrates the impact of SMOTE augmentation on the non-linear relationships between features. Minor deviations across the dataset indicate effective preservation of non-linear dependencies, demonstrating that SMOTE-like data generation successfully maintains the integrity of feature correlations



**Fig.6 Aggregate Impact Analysis on Correlation Structures: Original vs. SMOTE-Augmented Data.** This heatmap delineates the comprehensive differences in correlation coefficients between the original and the SMOTE-augmented dataset, quantifying the net impact of data augmentation on the linear relationships across all variables. Each cell represents the discrepancy in correlation for each pair of features, annotated to show specific differences. The color gradient of the heatmap, ranging from purple to green, indicates the magnitude of correlation changes, where purple suggests a decrease and green an increase in correlation relative to the original dataset. The subtle variations in most matrix values, especially for the final selected features, indicate that SMOTE effectively preserves the overarching correlation structure

The investigation below assesses the capacity of SMOTE to accurately mirror the distributional characteristics of original dataset. Utilizing the Kolmogorov-Smirnov (KS) and Anderson-Darling (AD) tests, the study quantitatively evaluates the similarity between original and SMOTE-augmented datasets. This analysis is crucial for validating the effectiveness of SMOTE in generating representative synthetic data, particularly in scenarios where expanding dataset size is essential for robust predictive modeling.

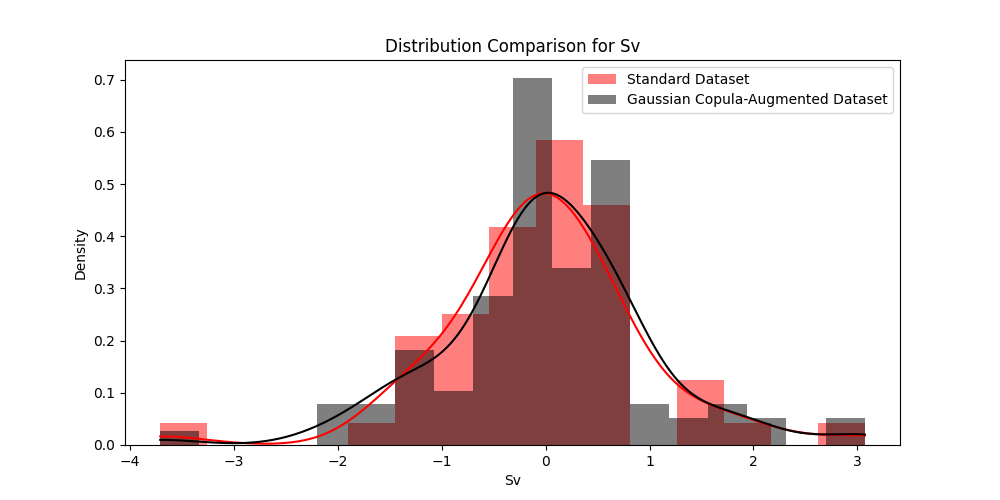
**Table 1 Statistical comparison of sample feature distributions between original and SMOTE-augmented datasets:** Analysis demonstrating that SMOTE effectively mirrors the original distributions

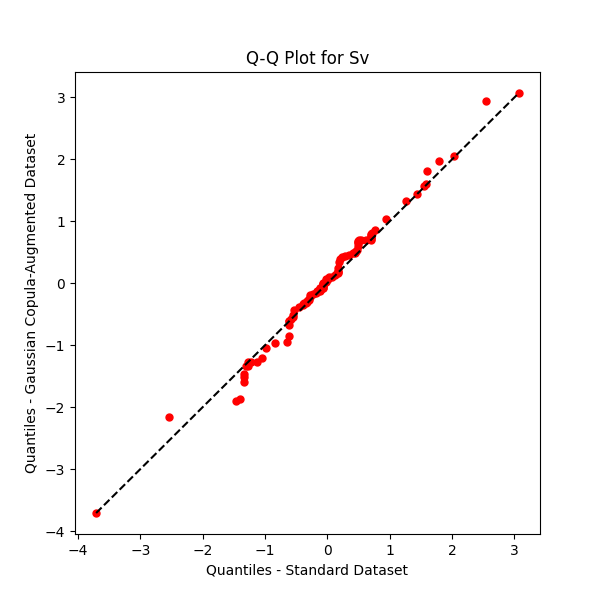
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Molecular Descriptor** | **KS Statistic** | **KS P-value** | **AD Statistic** | **AD Critical Values** | **AD Significance Level** |
| Sv | 0.1124 | 0.5487 | 0.1620 | [0.325, 1.226, 1.961, 2.718, 3.752, 4.592] | 0.25 |
| NumValenceElectrons | 0.1158 | 0.5105 | 0.2916 | [0.325, 1.226, 1.961, 2.718, 3.752, 4.592] | 0.25 |
| Kappa2 | 0.1537 | 0.1918 | 0.3101 | [0.325, 1.226, 1.961, 2.718, 3.752, 4.592] | 0.25 |

Notes:

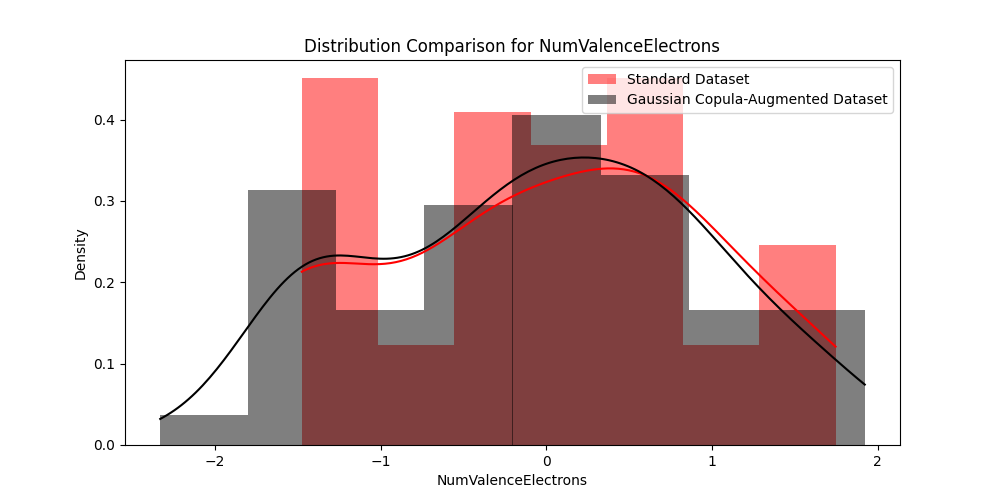
* KS Statistic and KS P-Value: The Kolmogorov-Smirnov (KS) test assesses whether two datasets originate from the same distribution. The KS statistic quantifies the maximum discrepancy between the cumulative distribution functions of the two datasets. The p-value represents the probability of observing the computed KS statistic, or one more extreme, assuming the null hypothesis of identical distributions is true. Thus, a higher p-value indicates a lower likelihood of falsely rejecting the null hypothesis, implying greater similarity between the compared distributions. A smaller KS statistic, which ranges from 0 to 1, and a high p-value (generally p > 0.05) suggest that the two datasets are from the same distribution with no statistically significant difference.
* AD Statistic and AD Critical Values: The Anderson-Darling test assesses the goodness-of-fit of the data to a standard distribution. It measures how well the empirical distribution of the data conforms to the standard distribution, with a focus on the tails of the distribution. The AD statistic is a numerical value that quantifies the discrepancy between the observed and original distribution patterns. The AD critical values are specific thresholds determined by the chosen significance level; if the AD statistic exceeds these critical values, the null hypothesis that the data follow the original distribution is rejected [56].
* AD Significance Level: In conducting the Anderson-Darling test, we set the significance level at 0.25, higher than the standard level of 0.05. This elevated threshold allows for a more lenient criterion in detecting differences between distributions, which does increase the likelihood of Type I errors—instances where the null hypothesis is incorrectly rejected. Nonetheless, this approach is strategically chosen to significantly reduce the risk of Type II errors, which occur when actual differences are not detected. Employing a higher threshold is particularly advantageous in scenarios where the consequences of missing a genuine effect are more severe than those associated with a false detection. This method is crucial in ensuring that any potential differences between the distributions, even subtle ones, are identified.

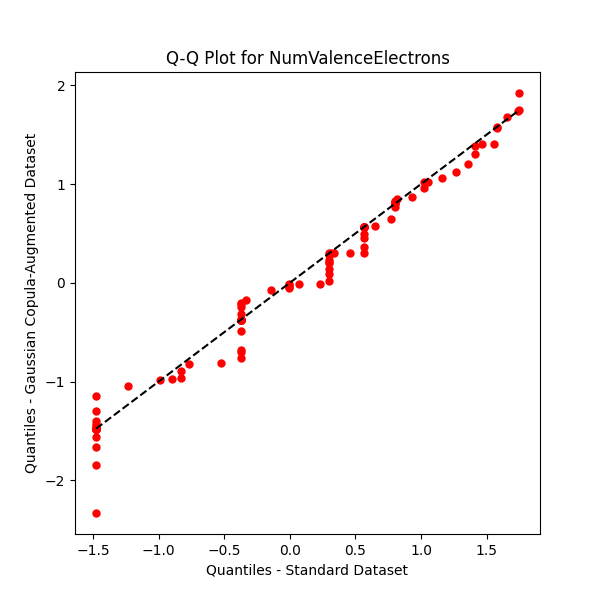
The following statistical investigation focuses on a Gaussian Copula-augmented dataset that includes both interpolated and extrapolated datapoints, extending beyond the scope of typical augmentation methods. By accurately capturing and replicating the joint distributions and inherent correlations of descriptors, the following analysis rigorously evaluates the fidelity of this augmented dataset, distinguished by its inclusion of extrapolated data points, in preserving the original data's statistical properties.

**a)**

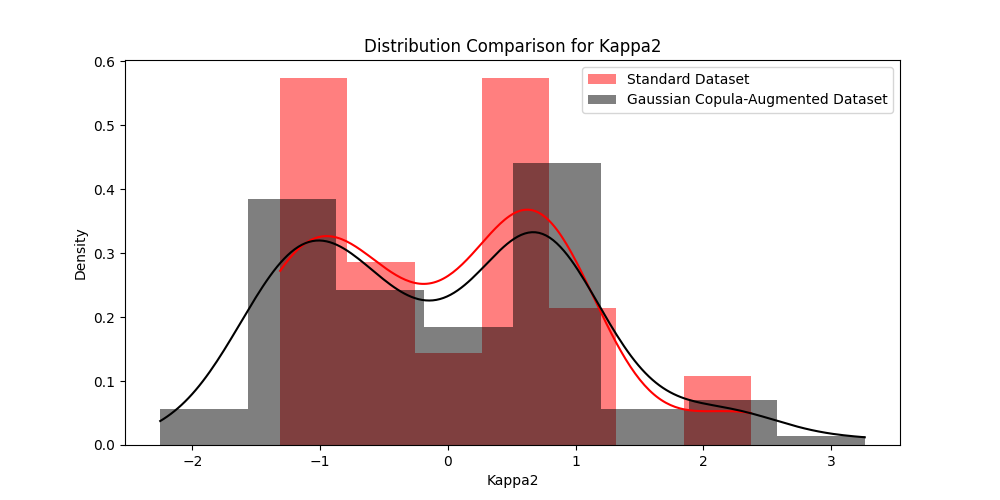
**b)**

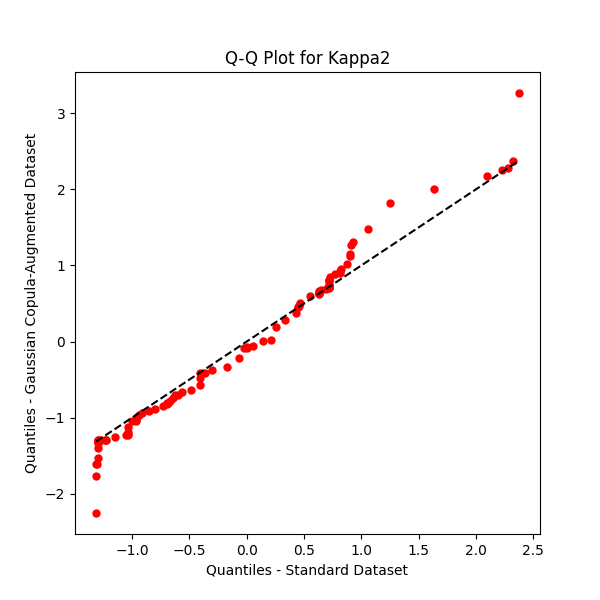
**Fig.7 The two figures rigorously assess the alignment in distributional characteristics of the descriptor "Sv" between a standard dataset and a Gaussian Copula-augmented counterpart.** Figure 7a, depicting both histograms and kernel density plots, illustrates a pronounced congruence in the central distribution regions between the two datasets, with any discrepancies in the tails being subtle. The Quantile-Quantile plot in Figure 7b reinforces this finding by displaying a predominantly precise overlay of quantiles along the diagonal reference line, indicative of a robust statistical concordance. Minor deviations from this line, observed at the distributional extremes, are statistically insignificant, underscoring the Gaussian Copula method's effectiveness in faithfully replicating the overall distribution of the original dataset

**a)**

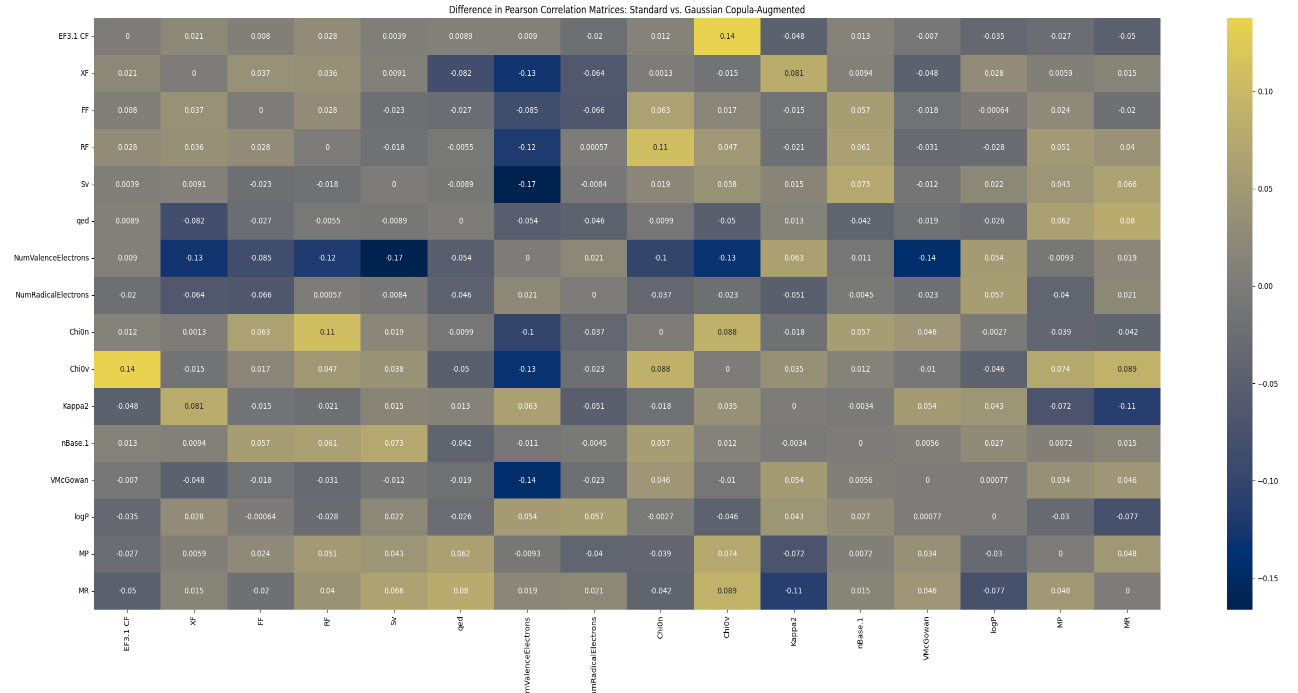
**b)**

**Fig.8 Comparative analysis of distributional characteristics for "NumValenceElectrons" between Standard and Gaussian Copula-Augmented datasets.** Panel a) presents a histogram with kernel density estimates comparing the distribution of the feature between the standard dataset (in red) and the Gaussian Copula-augmented dataset (in black). Panel b) is a Q-Q plot that visually assesses the quantile alignment of the same feature across these datasets. Both panels together provide a detailed evaluation of the Gaussian Copula method's effectiveness in replicating the original data distribution, highlighting a strong alignment in central tendencies and overall distribution shapes, with minor deviations at the distribution tails. These visualizations complement statistical tests, supporting the conclusion that the Copula method effectively simulates the distribution characteristics of the original dataset

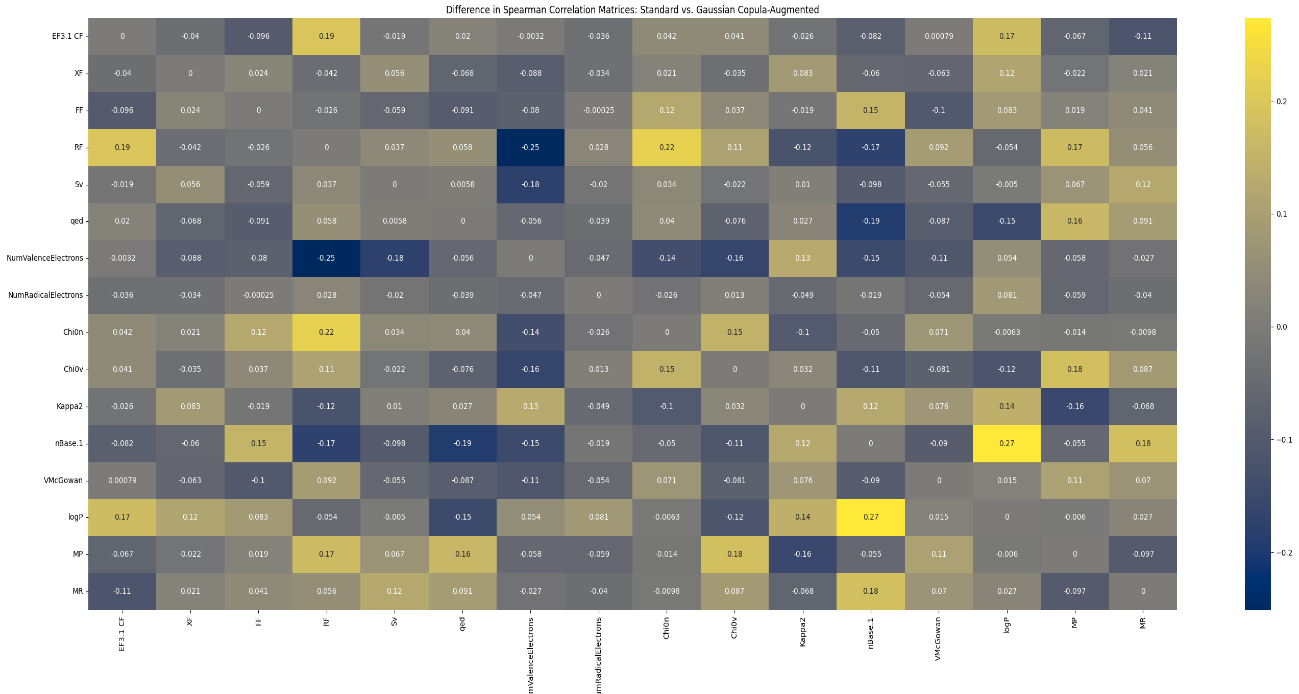
**a)**

**b)**

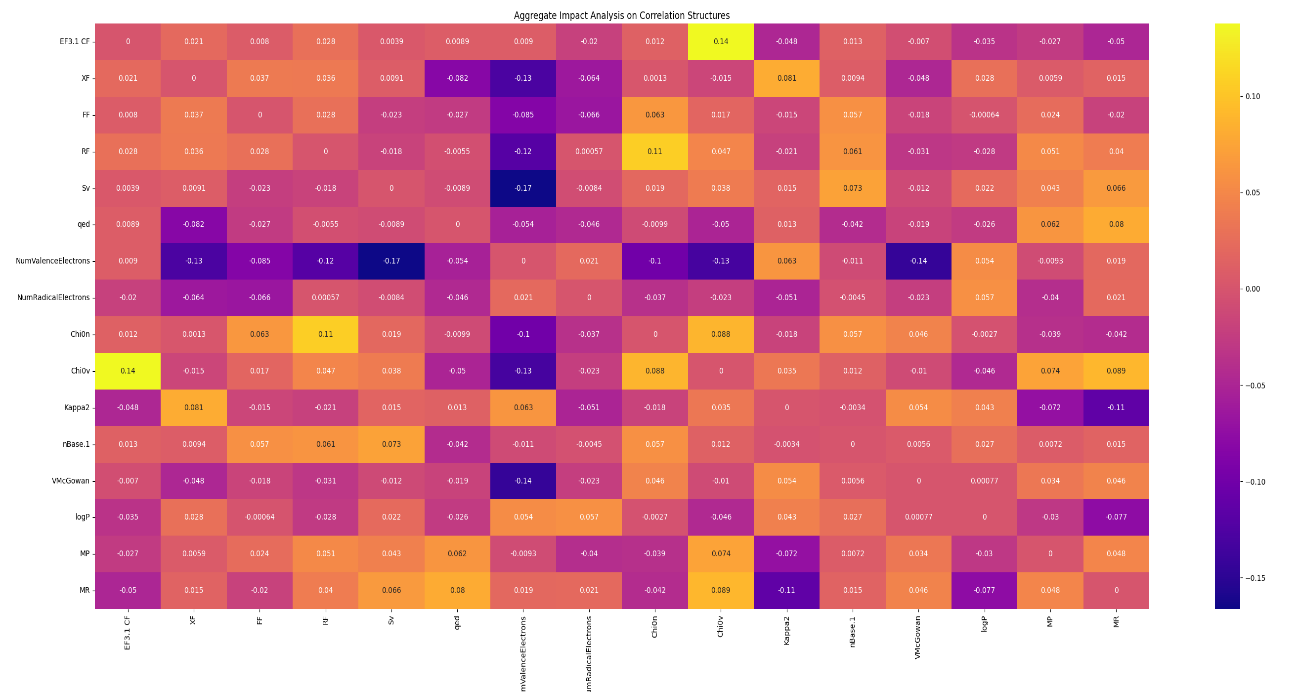
**Fig.9 The two figures collectively provide a comparative analysis of the distributional characteristics of "Kappa2" between the standard and Gaussian Copula-augmented datasets.** Figure 9a, which presents both histogram and kernel density estimates, reveals the core distributional alignment alongside subtle discrepancies in the extremities, suggestive of the augmented dataset's efficacy in replicating the central distribution while differing slightly in the distribution tails. Figure 9b employs the Quantile-Quantile plot to further validate these findings, illustrating a strong quantile alignment that corroborates the augmented dataset's capacity to mirror the fundamental statistical properties of the standard dataset, with minor variations observed at the distribution's boundaries. Together these visualizations demonstrate the Gaussian Copula method's efficacy in preserving the statistical integrity of the original dataset through interpolation and extrapolation, thereby validating its applicability for advanced data augmentation tasks



**Fig.10 Comparative analysis of Pearson Correlation Coefficient Differences between Standard and Gaussian Copula-Augmented datasets.** This heatmap visualizes the differences in Pearson correlation coefficients between the standard dataset and the Gaussian Copula-augmented dataset across the 15 descriptors. Each cell represents the difference in correlation for a pair of features, with positive values indicating a higher correlation in the standard dataset and negative values denoting higher correlations in the Gaussian Copula-augmented dataset. The color gradient, ranging from blue to yellow, illustrates the magnitude of these differences, with blue representing decreases and yellow representing increases relative to the standard dataset. The heatmap confirms the Gaussian Copula method's effectiveness in closely replicating the statistical properties of the original dataset, as evidenced by the uniformity in correlation coefficients across both datasets



**Fig.11 Comparative analysis of Spearman Correlation Coefficient Differences between Standard and Gaussian Copula-Augmented datasets.** Each cell represents the change in rank-order correlation for a pair of features, with the color scale ranging from blue to yellow, indicating decreases and increases in correlation, respectively, compared to the standard dataset. Values close to zero, shown in neutral colors, denote minor differences, while more saturated colors signify more substantial deviations. The majority of the cells exhibit subtle variations, especially for the eight descriptors, suggesting that the Gaussian Copula method finally preserves the non-linear relationships between features. This visual comparison underscores the method's capability to effectively mirror the original dataset's correlation structure in rank-order terms, highlighting its suitability for applications requiring the maintenance of inherent statistical dependencies within the data



**Fig.12 The heatmap provides an aggregate impact analysis on the correlation structures, comparing the differences between the original dataset and the Gaussian Copula-augmented dataset across the fifteen descriptors.** The color gradient in the heatmap, utilizing a "plasma" color map, visually represents the degree of difference in Pearson correlation coefficients, where each cell quantifies the discrepancy for a specific pair of descriptors. Positive values (shades of orange to yellow) indicate higher correlations in the standard dataset relative to the augmented one, while negative values (shades of purple) suggest higher correlations in the augmented dataset. The annotations within each cell provide precise numerical differences, enabling detailed scrutiny of the correlation changes. The numerical data, mainly showing small deviations close to zero, suggests that while there are some differences in the correlation structures between the datasets, these differences are minor. This demonstrates that the Gaussian Copula method, applied with extrapolation, has effectively preserved the original dataset's correlation structure, demonstrating its capability to effectively mirror the relational dynamics of the original dataset

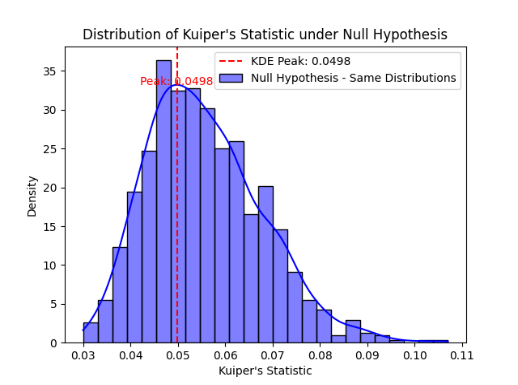
In evaluating the distributional similarities between original and Gaussian Copula-augmented datasets, for the regression task, Kuiper's test, complemented by the Cramér-von Mises test, emerges as the most fitting methodology. Kuiper's test effectively targets deviations across the entire cumulative distribution function, including both tails, making it ideal for our datasets where the Gaussian copula introduces complex dependencies and non-standard tail behaviors that the Anderson-Darling test cannot handle. This is because Kuiper's test is sensitive to deviations in both tails and the center of the distribution, whereas the KS test, though effective, is relatively less sensitive to deviations in the tails [55].

**Table 2 Detailed statistical analysis of features and target variable distributions in Original vs. Gaussian Copula-Augmented dataset**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Cramér-von Mises Statistic** | **Cramér-von Mises P-value** | **Kuiper's V Statistic** | **Kuiper's V P-value** |
| Sv | 0.0382 | 0.9500 | 0.1158 | 0.9950 |
| NumValenceElectrons | 0.0290 | 0.9841 | 0.1393 | 0.9457 |
| Kappa2 | 0.0367 | 0.9568 | 0.1402 | 0.9420 |
| *EF3.1 CF (Target Variable)* | *0.0423* | *0.9295* | *0.1333* | *0.9658* |

Notes:

* The Cramér-von Mises test meticulously assesses the goodness of fit between two empirical distribution functions by calculating the squared integrated differences across all data points. This test effectively measures the overall similarity between two distributions. The recorded statistics for the features range from 0.0290 to 0.0423, accompanied by p-values spanning from 0.9295 to 0.9841, suggesting no statistically significant deviation from the original distributions. These findings affirm the Gaussian Copula method’s fidelity in replicating the entire distributional structure of the original dataset, not just retaining specific values but ensuring overall distributional integrity. Typically, Cramér-von Mises Statistics’ values near zero, particularly values under the conventional threshold of 0.05 for non-significance, reflect a successful preservation of distributional characteristics [31], emphasizing the method's effectiveness in maintaining the integrity of data distributions.
* Kuiper's V test is employed to evaluate the distributional similarities between features in the original dataset and the Gaussian Copula-augmented. This test is specifically designed to measure the maximum discrepancies in the cumulative distribution functions at both the upper and lower tails. By doing so, Kuiper's V test excels in detecting deviations across the entire distribution range, including the tails, making it particularly effective for identifying differences in the distributional characteristics of the datasets. This test returned statistics ranging from 0.1158 to 0.1402, indicative of detectable differences. To contextualize these findings, null hypothesis simulations - assuming no intrinsic differences between datasets - produced Kuiper's statistics mostly within the 0.03 to 0.11 range, peaking near 0.05 (refer to **Fig. 13**). Despite the empirical Kuiper's statistics being higher than those from the simulations, suggesting distributional deviations especially at the tails, the corresponding high p-values (ranging from 0.9420 to 0.9950) indicate that these discrepancies do not reach statistical significance. Thus, while minor, observable deviations are revealed, particularly at the distribution tails, there is insufficient evidence to reject the hypothesis of distributional equivalence. These results demonstrate the efficacy of the Gaussian Copula method in maintaining the distributional characteristics throughout the augmentation process.



**Fig.13 Distribution of Kuiper's Statistic under the Null Hypothesis.** This histogram and overlaid kernel density estimate (KDE) illustrate the distribution of Kuiper's Statistic values generated from 1,000 simulations where both sample datasets are drawn from the same distribution. The KDE peak at 0.0498 indicates the most frequent Statistic value observed under the assumption of no significant differences between the datasets. This histogram provides a foundational reference, facilitating the quantitative assessment of deviations in the study’s empirical datasets relative to expected outcomes under the null hypothesis

***Procedure for Predicting the CF ECOTOX Class of Inorganic Elements Using the Developed Classification Model (Implemented in Python)***

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

import pandas as pd

import numpy as np

np.random.seed(1999999999)

# Load the model's training set. See it in Supplementary Material 1, under the tab "Classification Training Set"

my\_data\_path = r"C:\Users\Supplementary Material 1.xlsx"

my\_data = pd.read\_excel(my\_data\_path, sheet\_name="Classification Training Set")

# Prepare features and target for classification

X\_train = my\_data.drop("Class", axis=1)

y\_train = my\_data["Class"]

# Load the unseen dataset

unseen\_data\_path = r"C:\Users\Unseen dataset.xlsx"

unseen\_data = pd.read\_excel(unseen\_data\_path)

# Add an empty 'Class' column for consistency

unseen\_data["Class"] = pd.NA

# Initialize the classifier model

best\_model = RandomForestClassifier(random\_state=1999999999)

# Train the classifier

best\_model.fit(X\_train, y\_train)

# Predict the class for the unseen data (excluding the empty 'Class' column)

class\_predictions = best\_model.predict(unseen\_data.drop(columns=["Class"]))

# Print the predictions

print("Predicted Classes for the Unseen Dataset:")

print(class\_predictions)

### Procedure for Applying the Yeo-Johnson Transformation to New, Unseen Datapoints ###

# First, calculate all the molecular descriptors for the new, unseen elements using the four toolkits: PaDEL, RDKit, Mordred, and Pybel, and save the results in an Excel file.

# Then, apply the short code block below to the generated Excel file and export the new Excel file named "filtered\_untransformed\_dataset".

# Path to the dataset containing all molecular descriptors for the new unseen elements.

dataset\_with\_all\_descriptors\_path = r"C:\Users\Dataset\_with\_all\_descriptors.xlsx"

# Read the Excel file into a dataframe

produced\_dataset\_df = pd.read\_excel(dataset\_with\_all\_descriptors\_path)

# Remove columns where all rows have the same value

filtered\_dataset\_df = produced\_dataset\_df.loc[:, produced\_dataset\_df.nunique() != 1]

# Export the filtered dataframe to a new Excel file

filtered\_output\_path = r"C:\Users\Filtered\_untransformed\_dataset.xlsx"

filtered\_dataset\_df.to\_excel(filtered\_output\_path, index=False)

# Display message confirming the export to the Excel file

print(f"Filtered untransformed dataset saved to: {filtered\_output\_path}")

# Subsequently, to effectively apply the Yeo-Johnson transformation, first add the Filtered untransformed dataset to the dataset containing the 51 elements and the 190 molecular descriptors (**see Suppl. Mat. 1**), and name the new Excel file "final\_filtered\_untransformed\_dataset". Before doing this, remove the columns named "XF" and "FF" if these values are not available for the unseen elements, as these descriptors cannot be calculated using the toolkits mentioned earlier and require specific external data sources for their determination. Additionally, calculate the RF descriptor for the unseen elements using the straightforward method outlined in the paper.

# The Yeo-Johnson transformation

from sklearn.preprocessing import PowerTransformer

# Path to the filtered\_untransformed\_dataset

final\_filtered\_untransformed\_dataset\_path = r"C:\Users\Final\_filtered\_untransformed\_dataset.xlsx"

# Read the filtered untransformed dataset

final\_df = pd.read\_excel(final\_filtered\_untransformed\_dataset\_path)

# Apply the Yeo-Johnson power transform to descriptors

power = PowerTransformer(method='yeo-johnson')

transformed\_dataset = pd.DataFrame(

    power.fit\_transform(final\_df),

    columns=final\_df.columns)

# Remove the first 51 rows (from index 0 to 50 inclusive), representing the 51 elements from the EF v.3.1 database

transformed\_dataset = transformed\_dataset.iloc[51:]

# Keep only the eight molecular descriptors

columns\_to\_keep = ['RF', 'Sv', 'qed', 'NumValenceElectrons', 'Chi0v', 'Kappa2', 'VMcGowan', 'logP']

final\_transformed\_dataset = transformed\_dataset[columns\_to\_keep]

# Path to save the final transformed dataset

final\_output\_excel\_path = r"C:\Users\Final\_transformed\_dataset.xlsx"

# Export the dataframe to Excel

final\_transformed\_dataset.to\_excel(final\_output\_excel\_path, index=False)

# Display message confirming the export

print("Final transformed dataset exported to Excel successfully.")

### End of the Yeo-Johnson Transformation procedure

***Consolidated code for predicting Element classes and corresponding CF values***

The following code implements an integrated approach for predicting the classes of inorganic elements, followed by estimating their precise CF ΕCOTOX values. The results are compiled into an Excel file named "Predicted\_CF\_values.xlsx," ensuring a structured and clear presentation of the CF value predictions.

import pandas as pd

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeRegressor

import numpy as np

np.random.seed(1999999999)

# Load the model's training set

my\_data\_path = r"C:\Users\Supplementary Material 1.xlsx"

my\_data = pd.read\_excel(my\_data\_path, sheet\_name="Classification Training Set")

# Prepare features and target for classification

X\_train = my\_data.drop("Class", axis=1)

y\_train = my\_data["Class"]

# Load the unseen dataset

unseen\_data\_path = r"C:\Users\Unseen dataset.xlsx"

unseen\_data = pd.read\_excel(unseen\_data\_path)

# Add an empty 'Class' column for consistency

unseen\_data["Class"] = pd.NA

# Initialize the classifier model

best\_model = RandomForestClassifier(random\_state=1999999999)

# Train the classifier

best\_model.fit(X\_train, y\_train)

# Predict the class for the unseen data (excluding the empty 'Class' column)

class\_predictions = best\_model.predict(unseen\_data.drop(columns=["Class"]))

# Print the predictions

print("Predicted Classes for the Unseen Dataset:")

print(class\_predictions)

# Load the overall training dataset for the regression task. This dataset is located in Supplementary Material 1, under the tab named "Regression full set."

regression\_data\_path = r"C:\Users\Supplementary Material 1.xlsx"

df\_train = pd.read\_excel(regression\_data\_path, sheet\_name="Regression full set")

# Function to execute the regression model based on the class assigned to the new, unseen elements by the classification model

def run\_regression(df\_train, indices, unseen\_data, element\_name):

X\_train\_reg = df\_train.iloc[indices].drop(columns=["EF3.1 CF"])

y\_train\_reg = df\_train.iloc[indices]["EF3.1 CF"]

# The ml model

model = DecisionTreeRegressor(random\_state=1000000000)

# Fitting the model to the training data

model.fit(X\_train\_reg, y\_train\_reg)

# Predict on user's new datapoints

y\_unseen\_pred = model.predict(unseen\_data.drop(columns=["Class"]))

# Create the dataframe with the following specified structure

export\_data = pd.DataFrame({

"SMILES": [element\_name] \* len(y\_unseen\_pred),

"EF3.1 CF": y\_unseen\_pred})

return export\_data

# List to store all predictions

all\_predictions = []

# Set the column names to represent the specific elements in the unseen dataset e.g., Element\_1 == Magnesium

elements = ["[Element\_1]", "[Element\_2]"]

# Loop through each prediction and run the appropriate set of datapoints

for i, (pred\_class, element) in enumerate(zip(class\_predictions, elements)):

if pred\_class == 0:

indices = range(0, 100)

elif pred\_class == 1:

indices = range(100, 200)

elif pred\_class == 2:

indices = range(200, 300)

elif pred\_class == 3:

indices = range(300, 400)

elif pred\_class == 4:

indices = range(400, 500)

elif pred\_class == 5:

indices = range(500, 600)

result = run\_regression(df\_train, indices, unseen\_data.iloc[[i]], element)

all\_predictions.append(result)

# Combine all predictions into a single dataframe

combined\_predictions = pd.concat(all\_predictions, ignore\_index=True)

# Export the predictions to an Excel file

combined\_predictions.to\_excel(r"C:\Users\Predicted\_CF\_values.xlsx", index=False)