1 Introduction

With the advancement of Internet of Things (IoT) technology, we have the ability to collect a vast amount of time series data for various applications, which in turn increases the complexity of understanding the collected data. Processing and analyzing such large-scale data has become a crucial step in uncovering the underlying characteristics of these applications. In the field of data exploration, numerous methods have been developed to extract valuable information from time series data. However, interpreting the results produced by these solutions can often pose a challenge. To aid in the interpretation process, experts often rely on visual analysis tools [1]. Visual analytics [2] combines the computational capabilities of tools with human exploration and analysis skills to provide insights. Its objective is to integrate automated analysis methods with interactive visualizations, enabling users to extract potential information and gain knowledge from raw data. Time series data is commonly visualized as a time series graph, which proves beneficial when dealing with smaller datasets. However, when confronted with large time series data, tasks such as anomaly detection, recurring pattern extraction, clustering similar patterns, obtaining uncompressed overviews, or compressing overviews present significant challenges. To tackle the difficulties posed by big data, dimension reduction (DR) methods are employed. DR aims to transform data from a high-dimensional space to a lower-dimensional space while preserving its underlying structure. Current DR methods, such as t-distributed stochastic neighbor embedding (t-SNE) [3] and Uniform Manifold Approximation and Projection (UMAP) [4], are designed to generate visualizations that assist users in gaining insights from the collected data. Although these methods are powerful, the hyperparameters tuning is still challenging for users. The selection of suitable hyperparameters is critical as it determines the quality and usefulness of the visualization. Obtaining the most effective visualization often requires a tedious and time-consuming process of trial and error for users. The process of automatically determining the optimal hyperparameter value for the DR method involves two primary issues: (1) defining a suitable measure for selecting the best hyperparameter values and (2) efficiently searching within the predefined range to find this value. This work focuses on addressing the challenge of hyperparameter selection within a predefined range by evaluating the quality of visualizations. The proposed approach aims to achieve an efficient and accurate search for the best hyperparameter values. The contribution of this work is to conduct data exploration for a single time series by employing a group algorithm to search for cluster phenomena using rolling windows and visualization on two-dimensional (2D) images using the DR algorithm and the automatic hyperparameter adjustment method. It ultimately presents a comprehensive algorithm called DataTalk Visualization (DataTalk-V) with a custom function implementation.

The paper is organized as follows: Section 2 overviews the related studies; Section 3 proposes our visualization algorithm called DataTalk-V; Section 4 evaluates the performance of the proposed algorithm; Section 5 uses the soil Bacillus number prediction as an example to show the effect of DataTalk-V.
2 Related Works

In [5], the authors employed the rolling window method along with DR technology to visualize time series data. This conversion of data into a 2D visual map enables users to swiftly identify recurring patterns, label distribution, and detect anomalies. However, the authors did not provide information regarding the selection of hyperparameters for the DR techniques.

Grouping is an unsupervised learning technique that partitions a dataset into clusters without labels. Through grouping, we can identify and summarize intriguing patterns and correlations within the underlying data. One widely-used and fundamental grouping algorithm is k-means [6]. The k-means algorithm utilizes the mean to represent the centroid of each cluster, dividing the dataset of data points into clusters in a way that classifies each data to the cluster with the nearest mean distance.

In contrast to k-means, the Partitioning Around Medoids (PAM) algorithm [7] aims to identify k representative objects, known as medoids, from a given dataset. It then assigns each object to the nearest medoid to create clusters. A medoid is defined as the object with the smallest average distance to all other points in the dataset. Consequently, unlike the mean, medoids are always real members of the dataset.

The study presented in [8] introduces k-shape, an algorithm designed for partitioned clustering. K-Shape effectively determines the center of time series data while accounting for scaling and displacement invariance. In terms of performance, k-shape surpasses all other contemporary cutting, hierarchical clustering, and spectral clustering algorithms, except for PAM with Dynamic Time Warping (DTW) [9].

The study presented in [10] proposes the utilization of the Silhouette Coefficient (or Silhouette score) to evaluate the effectiveness of grouping based on the cohesion degrees within groups and the separation between groups. This approach also serves as a means to determine the optimal number of groups in the grouping algorithm.

In [4], the authors introduce UMAP, a dimensionality reduction (DR) technique rooted in robust mathematical theory. UMAP leverages theoretical frameworks of Riemannian geometry and algebraic topology. It surpasses t-SNE in visualization quality and retains a greater portion of the global structure while exhibiting lower time complexity.

In [11], the authors introduce the concept of the Constraint Preserving Score (CPS). The CPS is a score that takes into account the user’s constraint conditions and measures the extent to which these constraints are preserved in the results of the DR. CPS aims to differentiate the visualization effects achieved by the DR outcomes. However, this method relies on the user having a certain level of understanding of the data and providing the relationships and constraints among the data to effectively define the CPS. The authors also describe how Bayesian Optimization (BO) [12] can be leveraged to efficiently search for optimal hyperparameters, replacing traditional search-based methods.

The study presented in [13] introduces a semi-supervised DR method called Semi-Supervised Local Fisher Discriminant Analysis (SSLFDA). This method utilizes the grouping results obtained from a clustering algorithm as pseudo labels to extract discriminant information from unlabeled data. Experimental results demonstrate that SSLFDA achieves significantly superior classification performance on three hyperspectral imagery datasets compared to Fisher discriminant analysis, Local Fisher discriminant analysis, and Principal Component Analysis.

In [14], the authors employ the k-means score, k-NN Classifier top-1 accuracy score, and the Silhouette Coefficient to assess the effectiveness of the dimensionality reduction method. The k-means score and k-NN Classifier top-1 accuracy score evaluate the prediction labels in the low-dimensional embedding, while the Silhouette Coefficient measures the distribution of groups within the low-dimensional embedding.

The study discussed in [15] introduces a novel Cluster Validation Index (CVI) named Density-Based Clustering Validation Index (DBCV). Unlike most CVI methods that focus on measuring globular clusters, DBCV is specifically designed to identify high-density clusters separated by low-density regions. This index utilizes a new kernel density function to calculate the density of the data and evaluates the intra-cluster and inter-cluster density connectivity within the clustering results.

When dealing with mathematical models, sensitivity analysis entails ranking input variables by their importance, considering their intensity and correlation, and observing the resulting changes in the output. The variance-based sensitivity analysis method has been widely acknowledged in literature as a comprehensive and effective approach but is limited to univariate output models. In [16], a measure known as Aggregated Sobol’ Indices was proposed to quantify the sensitivity of input variables to the output of multivariate output models. In DataTalk-V, this measure is used to analyze the characteristics of soil Bacillus data in UMAP.

Traditional visualization methods rely on histograms for presenting one-dimensional data and scatterplots for displaying two-dimensional data. However, when the data dimensions exceed two, relying on multiple histograms or scatterplots side by side becomes insufficient for capturing the relationships among multiple dimensions. To address this limitation, a parallel coordinate diagram [17] can be utilized. By converting the scatter plot domain into two parallel and scalable axes within a parallel coordinate domain, a parallel coordinate plot enables the visualization of relationships among multiple dimensions. DataTalk-V incorporates parallel coordinate plots as one of its visualization methods.

In [18], we introduced a method that utilizes IoT and machine learning techniques to predict the number of Bacillus in soil. The soil data we used includes three turmeric fields (see R1, R2, and R3 in Fig. 1) covered with rice husks, as well as two other turmeric fields (P1 and P2 in Fig. 1) covered with peanuts. To analyze this soil data, specifically the soil sensor values and the number of Bacillus, we employ DataTalk-V as a visualization tool. Our aim was to explore the relationship between feature sensitivity and the predictive performance of a machine learning model.
3 DataTalk-V for Time Series Visualization

This section presents the DataTalk-V time series visualization algorithm. As illustrated in Fig. 2, DataTalk-V consists of four stages. The first stage extracts the features for pre-processing data. The second stage groups the data based on the silhouette score. The third stage reduces the dimensions by using the Constraint Preserving Score (CPS). The fourth stage tunes the hyperparameters based on Bayesian Optimization.

![Feature Extraction](image1.png)

Fig. 1. The five turmeric fields in the Bao Mountain Farm

3. A. Feature Extraction

In the feature extraction stage, we preprocess a time series dataset to produce a rolling window matrix that will serve as the input to the clustering stage. Let time series dataset \( D(t) \) be the input to DataTalk-V, which is expressed as

\[
D(t) = \{ X_t | 1 \leq t \leq T \}
\]

where \( X_t \) is the sample obtained at time \( t \). \( X_t \) has \( n \) features, and the \( \phi \)-th features is denoted as \( x_{t}^{\phi} \). Therefore, \( X_t \) is expressed as

\[
X_t = \{ x_{t}^{\phi} | 1 \leq \phi \leq n \}
\]

With a shift factor \( l \) and the step size \( s \) we can obtain a rolling window matrix \( E_{l}^{\phi}(t) \in \mathbb{R}^{n \times (l+1)} \) for \( \phi \), where \( 0 < l \leq t - t + 1 \), \( 0 \leq s \leq t - t - l \) and \( r = \frac{t - t - l}{s} \). From Eq. (2), \( E_{l}^{\phi}(t) \) is expressed as

\[
E_{l}^{\phi}(t) = \begin{bmatrix}
    x_{t}^{\phi} & x_{t+1}^{\phi} & \cdots & x_{t+l}^{\phi} \\
    x_{t+s}^{\phi} & x_{t+s+1}^{\phi} & \cdots & x_{t+s+l}^{\phi} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{r-t}^{\phi} & x_{r-t+1}^{\phi} & \cdots & x_{r-t+l}^{\phi}
\end{bmatrix}
\]  

We use Eq. (3) to define the rolling window matrix \( E_{l}(t) = [E_{l}^{1}(t) \ E_{l}^{2}(t) \ \cdots \ E_{l}^{n}(t)] \in \mathbb{R}^{n \times (l+1)n} \). Then \( E_{l}(t) \) is expressed as

\[
E_{l}(t) = \begin{bmatrix}
    x_{1}^{1} & x_{1+l}^{1} & \cdots & x_{1+n}^{1} \\
    x_{1+s}^{2} & x_{1+s+l}^{2} & \cdots & x_{1+s+n}^{2} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{r-t}^{n} & x_{r-t+1}^{n} & \cdots & x_{r-t+n}^{n}
\end{bmatrix}
\]  

For \( 1 \leq i \leq r \), let \( E_{l,i}(t) \) be the column vectors of \( E_{l}(t) \). Then Eq. (4) is rewritten as Eq. (5).

\[
E_{l}(t) = \begin{bmatrix}
    E_{l,1}(t) \\
    E_{l,2}(t) \\
    \vdots \\
    E_{l,r}(t)
\end{bmatrix}
\]

3. B. Clustering

In a clustering problem, we attempt to classify the data items of a dataset into \( h \) groups given that \( h_1 \leq h \leq h_H \), where \( h_1 \) and \( h_H \) are predefined by the user. In the clustering stage, the input given in Eq. (5) is the input to the clustering algorithm, and the output is a \( r \times 1 \) vector \( Y_{l}(t) \), where \( Y_{l,i}(t) \) is an integer representing the cluster of \( E_{l}(t) \) [11]. If we assign \( E_{l}(t) \) to one of \( h \) clusters, then \( 1 \leq Y_{l,i}(t) \leq h \) for \( 1 \leq i \leq r \). An important component of the clustering algorithm is the measure of distance between data points, especially in time series clustering. Time series clustering is primarily determined by classical clustering methods that replace the preset distance metric with a more suitable distance metric for time series data. Two time series clustering algorithms PAM-DTW [9] and k-shape [8] are used in DataTalk-V.

Both k-means [6] and PAM distribute data points into default \( h \) clusters, and then reshape the clusters using an iterative process. Each iteration includes an assignment step and a refinement step. In the assignment step, each data point is assigned to its nearest centroid or center cluster, as determined by the distance function. The refinement step updates the centroid or center of the cluster to reflect the change in the distance of the cluster members. The process converges when the cluster members do not change or the maximum number of iterations is reached. k-means aims to minimize the sum of the square Euclidean distances between data points and their centroid in each population. In contrast, PAM minimizes the sum of the paired distances of all data points in each swarm. PAM is known to be more resilient to noise and outliers compared to k-means, although it is more time-consuming.

The effectiveness of the clustering algorithm primarily relies on the selection of the distance function. Typically, the Euclidean distances (ED) are widely adopted for this purpose. When utilizing ED, the cluster center is determined through
the arithmetic mean. However, when dealing with time series data, the Dynamic Time Warping (DTW) measure is employed to compare distances between various time points in an aligned way. DataTalk-V utilizes PAM-DTW where the input is $E_i(t)$ defined in Eq. (5).

**Algorithm PAM-DTW**

**Input:** $E_i(t)$ is a $r \times m$ rolling window matrix, $h$ is the number of clusters.

**Output:** $Y_{i,h}(t)$ is a $r \times 1$ vector containing the assignment of $r$ instances to $h$ clusters.

1. Randomly choose $h$ data items as the initial medoids.
2. Assign each instance to the nearest medoid by using the DTW distance metric.
3. Calculate the cost which is the sum of DTW distances of all the instances to their nearest medoids.
4. Randomly select an instance.
5. Swap the medoid by the instance if the swap reduces the cost.
6. Repeat Steps 3-5 until the termination condition is met: either the $Y_{i,h}(t)$ values converge or the maximum number of iterations is reached.

The time complexities of PAM and DTW are $O(r^2 h)$ and $O(m^2)$, respectively. When DTW is used as a distance function of PAM, the time complexity is $O(m^2 r^2 h)$. In our example, PAM-DTW is selected when $r, m \leq 1000$.

To reduce the time complexity of PAM-DTW, DataTalk-V incorporates the k-shape clustering algorithm, which preserves the shape of time series data. The k-shape algorithm is a shape-based segmentation technique, maintaining the shape integrity of time series. Moreover, the distance metric utilized by k-shape remains unaffected by scaling and shifting. In order to expedite computations, k-shape employs a normalized version of the cross-correlation measure, which is a widely employed metric in signal and image processing. However, this measure is often disregarded in evaluation experiments for time series comparison problems due to the challenges associated with finding appropriate normalization and the potential inefficiency of cross-correlation implementations, which can be as slow as DTW. To address this, a distance metric known as shape-based distance (SBD) [8] can be effectively employed for domain-independent normalization, enabling efficient comparison of time series data. Experimental results have demonstrated that by combining k-shape with SBD, DataTalk-V achieves a computational speed improvement of two orders of magnitude compared to using PAM-DTW.

**Algorithm k-Shape**

**Input:** $E_i(t)$ is a $r \times m$ rolling window matrix, $h$ is the number of clusters.

**Output:** $Y_{i,h}(t)$ is a $r \times 1$ vector containing the assignment of $r$ instances to $h$ clusters.

1. Initialize the cluster assignment $Y_{i,h}(t)$ randomly.
2. Initialize $h$ zero vectors as cluster centroids.
3. For each cluster, find the new centroid by minimizing the sum of square cross-correlations between each instance and the centroid.
4. For each instance, compute the SBD for each new centroid.
5. Update $Y_{i,h}(t)$ according to the minimal SBD among those of the centroids.
6. Repeat Steps 3-5 until the termination condition is met: either the $Y_{i,h}(t)$ values converge or the maximum number of iterations is reached.

At the end of clustering (either PAM-DTW or k-Shape), $E_{i,t}(t)$ belongs to cluster $k$ if $Y_{i,h}(t) = k$. In these algorithms, the Silhouette score is a common and simple way to measure the quality of clustering results. For $1 \leq k \leq h$, let $G_{i,k}(t) = \{E_i(t) | Y_{i,h}(t) = k\}$ (6) and $|G_{i,k}(t)|$ is the size of $G_{i,k}(t)$. Denote the distance between $E_{i,t}(t)$ and $E_{i,j}(t)$ as $d(E_{i,t}(t), E_{i,j}(t))$. For every $E_{i,t}(t)$ assigned to cluster $k$ (i.e., $Y_{i,h}(t) = k$), we define the degree of dissimilarity (intra distance) $\delta_{d,k}$ within the cluster $G_{i,k}(t)$ and the degree of dissimilarity (nearest distance) $\delta_{n,k}$ between $E_{i,t}(t)$ and a cluster other than $k$ as

$$\delta_{d,k}(E_{i,t}(t)) = \sum_{E_{i,j}(t) \in G_{i,k}(t), j \neq i} \frac{d(E_{i,t}(t), E_{i,j}(t))}{|G_{i,k}(t)|} - 1$$

(7)

$$\delta_{n,k}(E_{i,t}(t)) = \min_{1 \leq l \neq k \leq h} \sum_{E_{i,m}(t) \notin G_{i,k}(t)} \frac{d(E_{i,t}(t), E_{i,m}(t))}{|G_{i,l}(t)|}$$

(8)

Eq. (8) says that it is more likely that $E_{i,t}(t)$ belongs to $G_{i,k}(t)$ (Eq. (6)) if $\delta_{d,k}(E_{i,t}(t))$ computed from Eq. (7) is small, and $E_{i,t}(t)$ is not likely to belong to other clusters if $\delta_{n,k}(E_{i,t}(t))$ computed in Eq. (8) is large. Therefore, for a specific $h$, we define the Silhouette score $S_k$ for cluster $k$ of $E_{i,t}(t)$ as

$$f_{S_k}(E_{i,t}(t), Y_{i,h}(t) = k)$$

$$\delta_{n,k}(E_{i,t}(t)) - \delta_{d,k}(E_{i,t}(t))$$

for $1 \leq k \leq h$ (9)

and for $h_k \leq h \leq h_H$, the silhouette score for $E_{i,t}(t)$ is defined using Eq. (9) as

$$S_k = F_{S_k}(E_{i,t}(t), h) = \frac{\sum_{i=1}^{r} f_{S_k}(E_{i,t}(t), Y_{i,t}(t))}{r}$$

(10)

Therefore, the optimal cluster number $h^*$ is computed using Eq. (10) as

$$h^* = \arg \max_{h_{i_h} \leq h \leq h_{max}} S_h$$

(11)

and $Y_{i,t}(t)$ computed based on $h^*$ is selected as the output of the clustering stage.
3. C. Dimension Reduction (DR)

To visualize $E_i(t)$ in two-dimensional plane, DataTalk-V uses UMAP [4] for DR. UMAP takes $E_i(t)$ as the input, and two hyperparameters are tuned to find the optimal results. The first hyperparameter $n_{neighbors}$ is the number of nearest neighbors, which is used to control the DR result more oriented to maintain the local structure or global structure. The second hyperparameter $min\_dist$ is used to control the tightness between the points of the DR result. DataTalk-V reduces the dimension $(t + 1)n$ of $E_i(t)$ to 2 to produce the embedding vector $Y_i(t) \in \mathbb{R}^{t \times 2}$, where

$$Y_i(t) = \begin{bmatrix} Y_{i,1}(t) \\ \vdots \\ Y_{i,r}(t) \end{bmatrix}$$

(12)

Let $\theta = (n_{neighbors}, min\_dist)$ be the hyperparameter pair of UMAP. Then based on the UMAP algorithm described in [4], Eq. (12) is obtained through

$$Y_i(t) = UMAP(E_i(t), \theta)$$

(13)

The K-fold cross-validation is introduced as the framework to accomplish the model selection task with the Bayesian Optimization method for automatic hyperparameter tuning of the pair $\theta$. The dataset is split into $K$ folds of training and validation sets (in the current DataTalk-V implementation, $K = 3$). By iterating Eq. (13) over the individual folds, there are $K$ UMAP models built with the $\theta$ values selected by the Bayesian Optimization under the given training sets. To avoid overfitting, the model selection among these K models are conducted according to a fused score $S_f$ to be elaborated later. Specifically, for each fold $\kappa$, the UMAP computes $Y_i^{(\kappa)}(t)$, and then DataTalk-V enters the Bayesian Optimization stage to find the best $\theta$ value denoted as $\theta^*(\kappa)$. Then we find the final $Y_i(t)$ using the following equation:

$$\theta^* = \arg\max_{1 \leq \kappa \leq K} S_f(Y_i^{(\kappa)}(t))$$

(14)

By substituting Eq. (14) into Eq. (13), we have

$$Y_i(t) = UMAP(E_i(t), \theta^*)$$

(15)

As we mentioned, to avoid overfitting, we develop a fused score $S_f$ by weighting three indexes: top-1 accuracy score $S_a$ of the k-NN Classifier, silhouette score $S_c$, and the Density-Based Clustering Validation Index (DBCV) $S_d$ [15].

The $S_d$ index indicates the quality of the clustering results. It serves as a measurement to the non-globular shape clusters. The validity index of a cluster $V_c$ is defined in equation 3.4 of [15]. It is built on top of Minimum Spanning Tree (MST) to decide the density separation among clusters. Specifically, the DBCV algorithm $F_{DBCV}$ computes this index over the $h$ clusters through the function $V_c$ and then calculates a weighted-average over them as the final index called $S_d$. That is, from Eq. (15)

$$S_d = F_{DBCV}(Y(t), Y_i(t)) = \sum_{k=1}^{h} \frac{|G_{i,k}(t)|}{|Y_i(t)|} V_c(G_{i,k}(t))$$

(16)

where

$$G_{i,k}(t) = \{Y_i(t) \mid Y_i(t) = k, 1 \leq i \leq r\}$$

(17)

DataTalk-V uses the top-1 accuracy score $S_a$ of k-NN Classifier as a measure for embedding, which is defined as follows:

$$S_a = \frac{TP + TN}{TP + FP + TN + FN}$$

(18)

where TP refers to the count of correctly predicted clusters, where the predicted and actual clusters align. TN represents the count of instances that do not belong to any cluster and are correctly identified as such. FP denotes the count of instances that are assigned to a cluster, but are not part of that cluster in reality. Lastly, FN indicates the count of instances that do not belong to any cluster, but are mistakenly predicted to belong to a cluster.

In addition to $E[S_a]$ (Eq. (10)) and $S_a$ (Eq. (18)), we also consider $S_d$ (Eq. (16)) to compensate for the inability to measure non-globular clustering. Both $E[S_a]$ and $S_d$ are normalized to $\hat{E}[S_a]$ and $\hat{S}_d$ such that $0 \leq \hat{E}[S_a], \hat{S}_d \leq 1$. That is, $\hat{E}[S_a] = \frac{\hat{E}[S_a] + 1}{2}$ and $\hat{S}_d = \frac{S_d + 1}{2}$. Then we mix these three indicators with two weights $\alpha$ and $\beta$ to obtain the indicator $S_f$ of model selection in DataTalk:

$$S_f = \alpha S_a + \beta \hat{E}[S_a] + (1 - \alpha - \beta) \hat{S}_d$$

(19)

where $0 \leq \alpha, \beta \leq 1$, and $\alpha + \beta \leq 1$. Then $S_f$ is used in Eq. (14) to find the final $Y_i(t)$ values.

3. D. Bayesian Optimization

DataTalk-V designs an objective function $F_D$ for Bayesian Optimization (BO) [12] to automatically tune the hyperparameter set $\theta = (n_{neighbors}, min\_dist)$ for UMAP.

$F_D$ uses the constraint preserving score $S_c$ to measure how well the constraint is preserved in embedding [11]. The score $S_c$ uses the cluster results $Y_i(t)$ as pseudo label formation constraint to measure the quality of UMAP embedding $Y_i(t)$ based on the values of hyperparameters $\theta$ [4]. Let similar set $Y_i^c(t)$ be the paired sets of $Y_i(t)$ belonging to the same clusters. Let $Y_i^p(t)$ be the paired sets of $Y_i(t)$ belonging to different clusters. Then the $S_c$ scores of $Y_i^c(t)$ and $Y_i^p(t)$ are computed by functions $F_3$ and $F_4$ defined in Eqs. (3) and (4) in [11], and the $S_c$ score for $Y_i(t)$ is defined in Eq. (5) in [11], which is expressed as

$$S_c = F_D(Y_i(t)) = \frac{1}{2} [F_3(Y_i^c(t)) + F_4(Y_i^p(t))]$$

(20)

In Eq. (20), the higher the $S_c$ score represents the more constraints retained and the better the result of DR. Bayesian Optimization is the model-based hyperparameter optimization that has been applied to hyperparameter tuning in many machine learning models, which can achieve better...
performance on the test set while requiring fewer iterations than random search. We use $S_c$ in Bayesian Optimization to automatically select the optimal UMAP hyperparameter set $\theta$. Specifically, we use the objective function $F_o$ to search for the optimal $S_c$ value in a range $\Theta_n$ for $n$ neighbors and the range $\Theta_m$ for $\text{min}_\text{dist}$. Following \cite{[4]} $\Theta_n = \left( \frac{2}{\sigma}, \frac{20}{\sigma} \right)$ and $\Theta_m = (0, 1.0)$ are selected. The Bayesian Optimization algorithm is described in the algorithm below, where $\theta[i]$ and $S_c[i]$ are the $\theta$ and the $S_c$ values computed in the $i$-th iteration. This algorithm uses the Gaussian Process (GP) regression model $\text{GP}(\Omega_\theta, \Omega_{S_c})$ and the Upper Confidence Bound $\text{UBC}(\theta_n, \theta_m, \text{GP}(\Omega_\theta, \Omega_{S_c}))$ to create the acquisition function, which is described in the Eq. (6) of \cite{[21]}. Specifically, the GP regression model takes a set of hyperparameter configurations $\Omega_\theta$ as inputs and the set $\Omega_{S_c}$ as the target values; The UBC takes the distribution generated by the GP model and the boundaries of the hyperparameters to sample the candidate hyperparameter pairs. The strategy combining the GP and UCB is referred as the GP-UCB method \cite{[21]}.

**Algorithm: Bayesian Optimization**

**Inputs:** $F_o$ (defined in Eq. (20)), $\theta_n$, $\theta_m$

**Outputs:** $\theta^*$

1. Initialize the first sampled hyperparameter pairs $\Omega_\theta \leftarrow \{ \theta \}$, where $\theta = (\theta_n \sim \text{Uniform}(\theta_n), \theta_m \sim \text{Uniform}(\theta_m))$
2. $\Omega_{S_c} \leftarrow \{ S_c \}$, where
   1. $Y_i(t) \leftarrow \text{UMAP}(E_i(t), \theta)$
   2. $S_c \leftarrow F_o(Y_i(t))$
3. $S_c^* \leftarrow \max x_{\Omega_{S_c}}$
4. $\theta^* \leftarrow \theta \left[ \arg \max x_{\Omega_{S_c}} \right]$
5. $f \leftarrow 1$
6. While $f = 1$ do
7. Fit a GP regression model $\text{GP}(\Omega_\theta, \Omega_{S_c})$
8. $\theta \leftarrow \text{UBC}(\theta_n, \theta_m, \text{GP}(\Omega_\theta, \Omega_{S_c}))$
9. $\Omega_\theta \leftarrow \Omega_\theta \cup \{ \theta \}$
10. $Y_i(t) \leftarrow \text{UMAP}(E_i(t), \theta)$
11. $S_c \leftarrow F_o(Y_i(t))$
12. $\Omega_{S_c} \leftarrow \Omega_{S_c} \cup \{ S_c \}$
13. If $S_c^* < S_c$ then $\theta^* \leftarrow \theta$; $S_c \leftarrow S_c$
14. If the termination condition is met: either the maximum number of iterations or the time budget is reached then $f \leftarrow 1$; otherwise $f \leftarrow 1$
15. Return $\theta^*$

4 Performance of Clustering and Dimension Reduction

It is clear that clustering and DR will lose information of the input features fed to the AI models. In this section, we show that by carefully design the $S_c$ function, DataTalk-V actually improves the performance of the $k$-NN model by reducing the noises in the original datasets. We use the largest and publicly available classified time series datasets in the UCR Time Series Classification Archive \cite{[20]}. This archive contains 128 sub-datasets with real and synthetic data across many different domains. Every data item in a data set has the same length. From the perspective of grouping, each data item belongs to only one category. Therefore, we can use three metrics $S_a$, $E(S_a)$ and $S_d$ to indicate which clustering each data belongs to. We can also measure the accuracy of the prediction and the degree of dispersion between clusters after dimensionality reduction. We used 50 datasets shown.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Size</th>
<th>Class</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>ArrowHead</td>
<td>211</td>
<td>3</td>
<td>251</td>
</tr>
<tr>
<td>Sensor</td>
<td>Car</td>
<td>120</td>
<td>4</td>
<td>577</td>
</tr>
<tr>
<td>Simulated</td>
<td>CBF</td>
<td>930</td>
<td>3</td>
<td>128</td>
</tr>
<tr>
<td>Device</td>
<td>Computers</td>
<td>500</td>
<td>12</td>
<td>725</td>
</tr>
<tr>
<td>Motion</td>
<td>CricketX</td>
<td>780</td>
<td>12</td>
<td>300</td>
</tr>
<tr>
<td>Motion</td>
<td>CricketZ</td>
<td>780</td>
<td>12</td>
<td>300</td>
</tr>
<tr>
<td>Image</td>
<td>DiatomSizeReduction</td>
<td>322</td>
<td>4</td>
<td>345</td>
</tr>
<tr>
<td>Image</td>
<td>DistalPhalanxOutlineAgeGroup</td>
<td>539</td>
<td>3</td>
<td>80</td>
</tr>
<tr>
<td>Image</td>
<td>DistalPhalanxOutlineCorrect</td>
<td>876</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>Image</td>
<td>DistalPhalanxTW</td>
<td>539</td>
<td>6</td>
<td>80</td>
</tr>
<tr>
<td>ECG</td>
<td>ECGFiveDays</td>
<td>884</td>
<td>2</td>
<td>136</td>
</tr>
<tr>
<td>Image</td>
<td>FaceFour</td>
<td>112</td>
<td>4</td>
<td>350</td>
</tr>
<tr>
<td>Sensor</td>
<td>MoteStrain</td>
<td>127</td>
<td>2</td>
<td>84</td>
</tr>
<tr>
<td>Image</td>
<td>Fish</td>
<td>250</td>
<td>7</td>
<td>463</td>
</tr>
<tr>
<td>Image</td>
<td>DistalPhalanxTW</td>
<td>539</td>
<td>6</td>
<td>80</td>
</tr>
<tr>
<td>Motion</td>
<td>GunPoint</td>
<td>200</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>Motion</td>
<td>GunPointAgeSpan</td>
<td>451</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>Motion</td>
<td>GunPointMaleVersusFemale</td>
<td>451</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>Spectro</td>
<td>Ham</td>
<td>214</td>
<td>2</td>
<td>431</td>
</tr>
<tr>
<td>Motion</td>
<td>Haptics</td>
<td>463</td>
<td>5</td>
<td>1092</td>
</tr>
<tr>
<td>Image</td>
<td>Herring</td>
<td>128</td>
<td>2</td>
<td>512</td>
</tr>
<tr>
<td>Sensor</td>
<td>InsectWingbeatSound</td>
<td>220</td>
<td>11</td>
<td>256</td>
</tr>
<tr>
<td>Sensor</td>
<td>ItalyPowerDemand</td>
<td>109</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>Device</td>
<td>LargeKitchenAppliances</td>
<td>720</td>
<td>3</td>
<td>720</td>
</tr>
<tr>
<td>Sensor</td>
<td>Lightning2</td>
<td>121</td>
<td>2</td>
<td>637</td>
</tr>
<tr>
<td>Sensor</td>
<td>Lightning7</td>
<td>143</td>
<td>7</td>
<td>319</td>
</tr>
<tr>
<td>Spectro</td>
<td>Meat</td>
<td>250</td>
<td>7</td>
<td>463</td>
</tr>
<tr>
<td>Image</td>
<td>MedicalImages</td>
<td>114</td>
<td>10</td>
<td>99</td>
</tr>
<tr>
<td>Image</td>
<td>MiddlePhalanxOutlineCorrect</td>
<td>891</td>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>Image</td>
<td>MiddlePhalanxTW</td>
<td>553</td>
<td>6</td>
<td>80</td>
</tr>
<tr>
<td>Image</td>
<td>OSULeaf</td>
<td>442</td>
<td>6</td>
<td>427</td>
</tr>
<tr>
<td>Sensor</td>
<td>Plane</td>
<td>210</td>
<td>7</td>
<td>144</td>
</tr>
<tr>
<td>Proximal</td>
<td>PhalanxOutlineAgeGroup</td>
<td>605</td>
<td>3</td>
<td>80</td>
</tr>
<tr>
<td>Refrigeration</td>
<td>Devices</td>
<td>720</td>
<td>3</td>
<td>720</td>
</tr>
<tr>
<td>ScreenType</td>
<td></td>
<td>720</td>
<td>3</td>
<td>720</td>
</tr>
<tr>
<td>SmallKitchenAppliances</td>
<td></td>
<td>720</td>
<td>3</td>
<td>720</td>
</tr>
<tr>
<td>SonyAI</td>
<td>RobotSurface1</td>
<td>621</td>
<td>2</td>
<td>70</td>
</tr>
<tr>
<td>SonyAI</td>
<td>RobotSurface2</td>
<td>980</td>
<td>2</td>
<td>65</td>
</tr>
<tr>
<td>SyntheticControl</td>
<td></td>
<td>600</td>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>ToeSegmentation1</td>
<td></td>
<td>268</td>
<td>2</td>
<td>277</td>
</tr>
<tr>
<td>ToeSegmentation2</td>
<td></td>
<td>166</td>
<td>2</td>
<td>242</td>
</tr>
<tr>
<td>BME</td>
<td></td>
<td>180</td>
<td>3</td>
<td>128</td>
</tr>
<tr>
<td>ShapeletSim</td>
<td></td>
<td>200</td>
<td>2</td>
<td>500</td>
</tr>
<tr>
<td>FreezerRegularTrain</td>
<td></td>
<td>300</td>
<td>2</td>
<td>301</td>
</tr>
<tr>
<td>FreezerSmallTrain</td>
<td></td>
<td>287</td>
<td>2</td>
<td>301</td>
</tr>
<tr>
<td>PowerCons</td>
<td></td>
<td>360</td>
<td>2</td>
<td>144</td>
</tr>
<tr>
<td>SmoothSubspace</td>
<td></td>
<td>300</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>UMD</td>
<td></td>
<td>180</td>
<td>3</td>
<td>150</td>
</tr>
</tbody>
</table>

The ‘size’ column ($r$) in Table 1 represents the total number of data points used in our experiment, which includes both the training and testing sets. DataTalk-V keeps the ratio of about 2:1 for training and
testing datasets. The ‘class’ column in the Table 1 denotes the number of classifications in each problem setting. The ‘Length’ column (\( l \)) denotes the time series length for an instance in a dataset. In the URC Time Series Archive [20], the length of each sequence is regularized into the consistent number in a dataset. All the time series data in this archive are one-dimensional time series. (This fact implies that \( n \) is 1.) In our experiment setting, the shift factor \( l \) is set to \( \tau - 1 \) for each dataset (In the car example, \( \tau = 577 \)). For the Car dataset example, \( E_i(t) \in \mathbb{R}^{120 \times 577} \) from Eqs. (4) and (5).

We conduct four sets of experiments: Experiment 1 is the baseline scenario that executes \( k \)-NN classification without clustering and DR; Experiment 2 performs UMAP with default hyperparameters to reduce the number of input features before \( k \)-NN is executed; Experiment 3 performs the UMAP with the pseudo label \( Y_i(t) \) produced by the clustering algorithm before \( k \)-NN is executed; and Experiment 4 performs the complete DataTalk-V stages (clustering, DR and Bayesian Optimization) with various \( \alpha \) and \( \beta \) setups of the \( S \) function (Eq. (19)), where \( \alpha = 0.5 \) and \( \beta = 0.25 \) in Experiment 4.1, \( \alpha = 0.33 \) and \( \beta = 0.33 \) in Experiment 4.2, \( \alpha = 0.4 \) and \( \beta = 0.2 \) in Experiment 4.3, and \( \alpha = 0.8 \) and \( \beta = 0.1 \) in Experiment 4.4.

Our study indicates that Experiment 4.2 (when \( \alpha = 0.33 \) and \( \beta = 0.33 \)) the performance of \( k \)-NN is better than other experiments. Specifically, its \( S_{\alpha} \) performance is better than other setups for 17 of the 50 UCR datasets and is better than the second best experiment by 1.75%. For the \( S_{\beta} \) performance, experiment 2 is better than other setups for 11 UCR datasets and is better than the second best experiment by 36.41%.

The above experiments show that the two-dimensional data reduced by DataTalk-V not only assists with data visualization but also improves the prediction accuracy \( S_{\alpha} \) of \( k \)-NN.

### 5 Visualization of Soil Sensor Data

This section shows visualization of soil sensor data collected in our previous work [18]. Over a nine-month period of turmeric cultivation in the Bao Mountain farm (as depicted in Fig. 1), we meticulously gathered soil sensor values every 15 seconds, resulting in a total of 1,083,370 samples. Each sample consisted of four sets of sensor data, namely: soil temperature (Temp), soil humidity (H), pH, and electrical conductivity (EC). To generate features for the AI model, we preprocessed the monthly mean (\( \text{Temp}_{\text{mean}}, H_{\text{mean}}, pH_{\text{mean}}, EC_{\text{mean}} \)) and monthly standard deviation (\( \text{Temp}_{\text{std}}, H_{\text{std}}, pH_{\text{std}}, EC_{\text{std}} \)) of the four sensors. In our previous study, we trained the machine learning model using the number of Bacillus (B) for the first five months. We used a voting regressor with Multilayer Perceptron (MLP) as the base estimators, and made predictions for the next four months [18]. This paper utilizes a total of 45 monthly averages and monthly standard deviations obtained from four soil sensors in five fields (P1, P2, and R1-R3 as shown in Fig. 1), resulting in eight features per data for the rolling window matrix \( E_i(t) \in \mathbb{R}^{45 \times 58} \), where \( l = 1, n = 8 \) and \( \tau = 9 \).

Following dimension reduction and clustering in DataTalk-V, the distribution of \( E_i(t) \) is displayed on the scatter plot in Fig. 3. The legend on the right side of the graph is divided into five colors, where a solid circle represents the members of the cluster and a hollow circle represents the center of the cluster. The 45 data points are clearly separated into five distinct clusters, and the clustering effect is highly significant. Each cluster is highly concentrated, and the data points within each cluster overlap and are indistinguishable. DataTalk-V provides an interactive graphical user interface, which allows users to zoom in or out on selected areas to observe more subtle structural changes.

Fig. 4 depicts the clusters plotted against the months, clearly indicating the appropriateness of the features and data preprocessing methods used for feature extraction in the AI model. The UMAP algorithm, which is better than linear methods in preserving the global relationships of data in low dimensional space, has been employed for this purpose. With successful grouping, these clusters are capable of pointing out the properties of data in higher dimensional space. By treating cluster movement as instances of discrete-state processes, we can observe the dynamic transitions of farm fields P over time. Initially, these fields transition from the blue cluster to the red cluster, then progress to the green cluster, and finally reach the purple cluster. Farm fields R, on the other hand, may temporarily visit an orange cluster before transitioning to the green cluster. These plotted patterns indicate that with the utilization of suitable data preprocessing techniques, we have successfully captured the trends in Bacillus' fluctuation accurately. Fig. 3 shows that...
the red cluster and green cluster are the closest in distance, indicating that these two soil states have the highest similarity. In Fig. 4, their states are also interchanged. The blue cluster and pink cluster are the farthest apart in distance in Fig. 3, indicating that these two soil states have the lowest similarity. They also undergo the most state transitions in Fig. 4.

6 Conclusions

The visualization of time series data plays a crucial role in aiding researchers to comprehend the inherent characteristics of the data and extract valuable insights and knowledge. Currently, one common approach for visualizing such data is to utilize dimensionality reduction (DR) techniques. However, a major challenge lies in the selection of an appropriate clustering algorithm and determining the optimal hyperparameters for dimensionality reduction. In order to address this challenge, we proposed DataTalk-V for time series visualization that combines clustering and DR methodologies.

DataTalk-V efficiently reduces high-dimensional data to a two-dimensional representation, offering clear and transparent visualization of the data. The novelty of our approach lies in two aspects. Firstly, DataTalk-V performs clustering as a preprocessing step on the high-dimensional data prior to applying dimensionality reduction. Secondly, DataTalk-V cleverly employs a cost function within Bayesian Optimization to effectively fine-tune the hyperparameters for the DR process. We showed that by carefully design the $S_f$ function, DataTalk-V actually improves the performance of the $k$-NN model by reducing the noises in the original datasets. We use Bacillus number prediction in smart turmeric farms as an example to show that scatter plot of soil sensor data and the soil data plotted in time by the classification status greatly assist the researchers to understand the complicated relationship among the number of Bacillus, the temperature, the humidity, the pH values and the electrical conductivity.

References


