Towards Visual Exploration of Large Temporal Datasets

Mohammed Ali, Mark W. Jones, Xianghua Xie and Mark Williams

July 16, 2018

Towards Visual Exploration of Large Temporal Datasets

M. Ali, M. W. Jones, X. Xie
Computer Science
Swansea University
Swansea, UK
mabood@kku.edu.sa
m.w.jones@swansea.ac.uk
x.xie@swansea.ac.uk

E. M. Williams
Faculty of Life Sciences and Education
University of South Wales
Pontypridd, UK
mark.williams@southwales.ac.uk

***Abstract*—Visual analytics for time series data has received considerable attention in previous literature, and different approaches have been developed to understand the characteristics of the data and to obtain meaningful information. Visualizing, analyzing and presenting large temporal datasets are important tasks to understand, navigate and explore such data. One-dimensional time-series charts are usually used to visualize time series data but if the dataset contains multiple time series with a large number of observations a high degree of overlap will occur which may obscure important information. This problem has become a vital challenge in many domains such as finance, biological systems, and meteorology. The need for analyzing and exploring large time-series data led researchers to develop various interactive visualization tools and analytical algorithms which aim to give insight into the data, and most of them either focus on a small number of tasks or a specific domain. We propose a visual analytics system and approach which aims to visualize, analyze, present and explore large temporal datasets. Our approach consists of three main stages which are preprocessing, dimensionality reduction, and visual exploration. It assists with finding the interesting features in the data which are often obscured in the line chart or the visual compression that is required to render the large datasets on a small screen. Also, it helps to obtain an overview of the entire dataset and track changes over time. Moreover, it enables the user to detect clusters and outliers and observe the transitions between data. The juxtaposed views are used to visualize and interact both with raw time series data and projection data. Different time series datasets are deployed on our system, and we demonstrate the utility and evaluate the results using a case study with two different datasets which show the effectiveness of our system.**

***Index Terms*—Time series data, Visual analytics, Time series graphs, Principal Component Analysis (PCA), 2D Projection, Clusters, Exploration**

I. Introduction

Due to an ever-increasing amount of time series data and the complexities involved with analyzing and understanding them in practice, revealing meaningful insights and knowledge from the shape of data has long been an active area of research. The processing and analyzing of such data require particular tasks and methods to support effective analysis. Besides that, visualization and interaction techniques are essential. Visual analytics is at the core of dealing with huge amounts of information by combining the enormous processing power and storage capacity of computers with the flexibility, creativity, and domain expertise of humans through interactive visual interfaces.

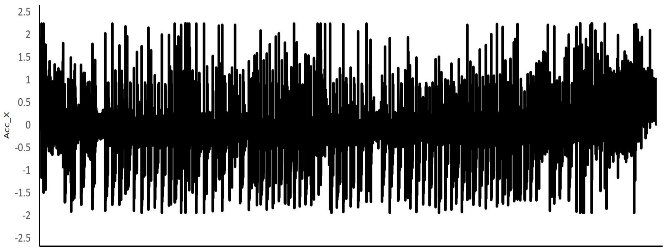


Fig. 1. 1D Line Chart with 173,256 data elements

The most common form of representing time-series are line plots which link the data points with a line that illustrates their temporal relation. However, one of the biggest challenges in time series visualization is getting an overview of a compressed or uncompressed line graph with the goal of gaining a better understanding of how relationships change over time, how information spreads, where clusters occur, where the common patterns occur, etc (Figure 1). It is difficult for the user to relate sequences of data that have long periods, particularly when they are a long temporal distance apart in the time-series data. Time series graphs are effective when dealing with a small data space, but performing common tasks on large data becomes more challenging. Many interaction techniques have been introduced to tackle that issue for large data, which will be reviewed in the related work, but most of those works focus on the interaction techniques while analysis techniques have been given less attention.

We propose a method of reducing every sliding window in a time series graph to a point based on a dimension reduction method and accelerating the dimension reduction using the property that we are using a sliding window on the time-series. This can ultimately present an overview of the whole-time series graph in one image. Consequently, selecting any points could give the user reasons why they are similar or

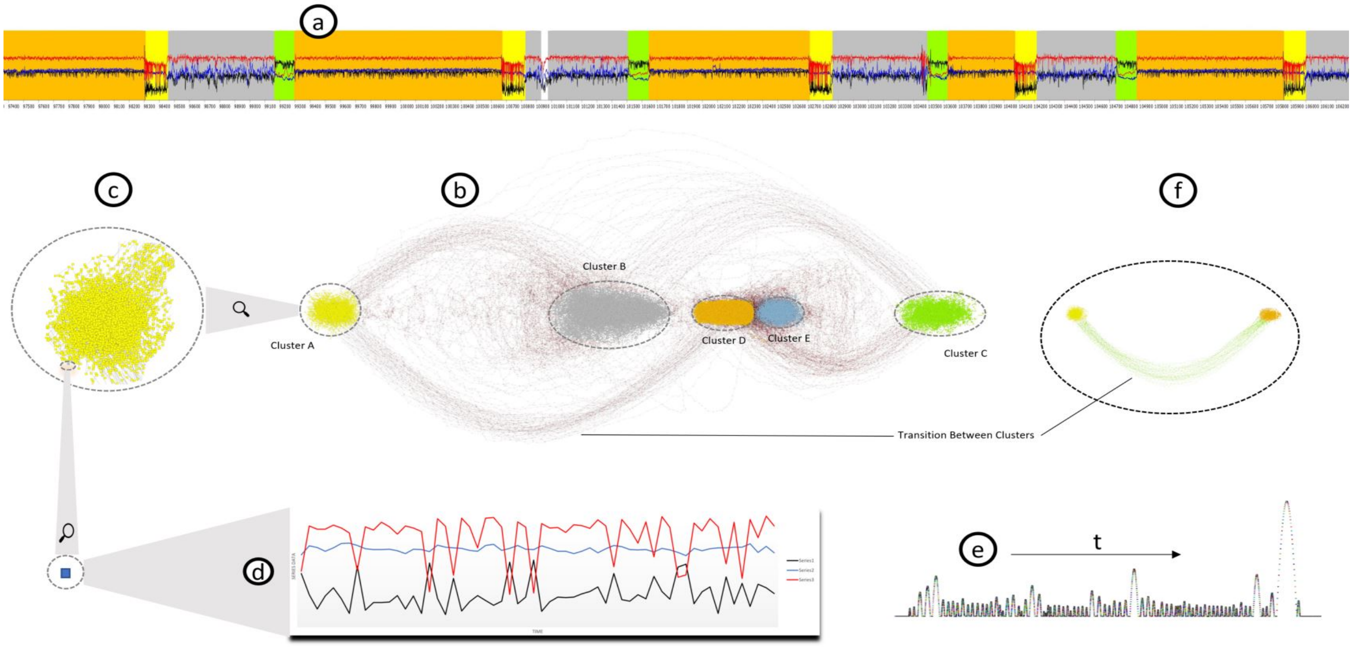


Fig. 2. The overview of the system. Exploration of five clusters and transactions between them after applying our approach on the time series data collected by a sensor from a Cormorant bird where different colors express different clusters, e.g. cluster A is one behavior (diving) which is concentrated in the same area. (a) time series graph for raw time series data, (b) connected scatter plot for the data after projection showing five clusters (Descent Phase of Dive (cluster A), Bottom Phase of Dive (cluster B), Ascent Phase of Dive (cluster C), Surface Swimming (cluster D), and Flight (cluster E)), (c) zooming cluster A, (d) drawing line chart for the selected data point, (e) time vs. first principal component (f) showing the transitions between cluster D to cluster A.

dissimilar, where clusters occur, and how the relationships between points develop, which are the interaction challenges that we tackle. The goal of this work is to provide a visual analytics system that assists users to understand and visualize time series graph simultaneously with the connected scatter plot that represents the whole dataset after the projection process (Figure 2). Furthermore, one of the important goals in our approach is to clearly show how the shape of data evolves over time, thus, researchers will be able to observe and understand the phenomena or behavior that occurs when comparing it over time.

This paper also addresses how visual analytics systems support automated clustering for a real-world problem assisting the user to understand the data, and, gaining insight in terms of clustering for instance, this could include how much the data changes within each cluster, which clusters are close to or distinct from each other, and where the most representative or relevant transitions have occurred between clusters in the phenomenon under consideration.

Our system does not require any templates for the matching process to take place. We utilize visualization and interaction techniques to search for matching patterns, which involve a user in the loop for checking, analyzing, and understanding. The main contributions of our work are:
1- Introducing a visual analytics system and approach that effectively depicts large time series data, facilitating the user to see, explore, trace, and understand large amounts of information from time series graph through a 2D connected scatter plot which summarizes the whole data on one image.
2- Showing the visual analysis capability of our system, which leads to a better understanding of large and complex temporal data and identifying underlying patterns, clusters, outliers, and transitions between them after projection.
3- Improving the PCA computation utilizing the stationary mean which executes on large datasets in real time to provide an interactive application.
4- Providing real-world use cases using two different time series datasets.

The approach consists of three essential steps:
1) Preprocessing; 2) Feature extraction and projection; 3) Visual Exploration.

The rest of this paper is organized as follows. In section 2, the related works are presented. Methodology is discussed in section 3. In section 4, two case studies are given. Finally, we conclude our findings in section 5.

II. Related work

In recent decades, many visual analytics systems that embed multiple visualization and interaction techniques have been proposed to deal with time series data. Supporting users in the discovery of potentially interesting patterns is one of the essential concepts in this domain by taking the advantage of the human ability to visually reveal and assess such patterns. A comprehensive overview is provided by Aigner et al. [1] which discusses and reviews an in-depth visualization of time-

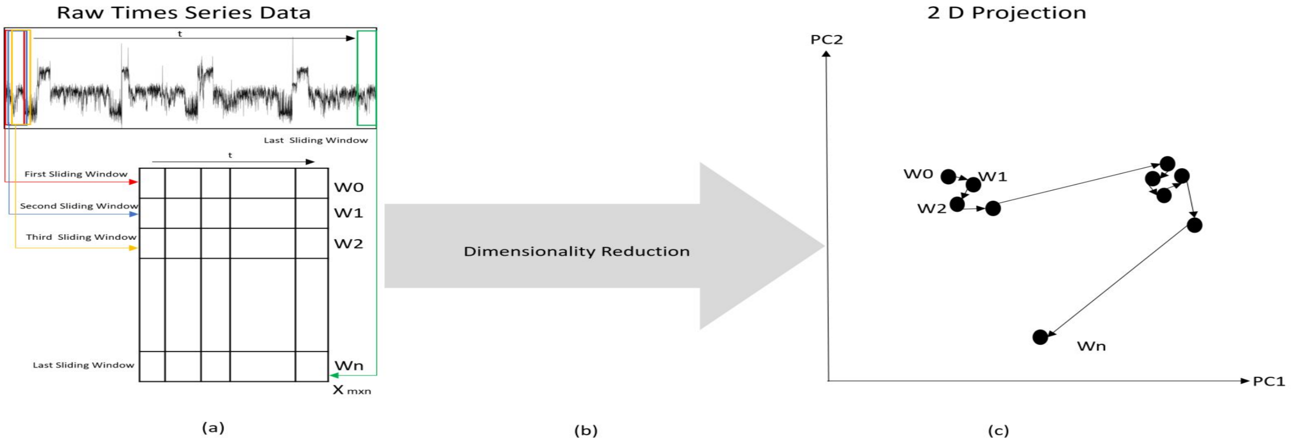


Fig. 3. Our system approach which consists of three major steps. (a) preprocessing, (b) dimensionality reductions, and (c) visual exploration.

oriented data illustrating with numerous examples and present a structured survey of existing techniques for visualizing and interacting with time-oriented data. In this section, we discuss only methods and systems that are pertinent to this paper.

To visualize motifs, Lin et al. [20] present VizTree, a visualization based on augmenting suffix trees which transforms a large time series into a symbolic representation. It allows mining and anomaly detection over large time series data. Time series data is converted to a symbolic representation (SAX), which is utilized to build a suffix tree that encapsulates the local and global structure of underlying time-series. Both tree and line charts are used to link different pieces of information. Ordonez et al. [24] append radial representations to their line chart for simplifying the motif analysis process. They demand user domain knowledge and interactions on the tree to understand a motif.

TimeSearcher 2 [6] is introduced which is the extension of TimeSearcher [10] for pattern discovery through **query-by-example** in the time-series data to find similar occurrences. Filtering is used to reduce the size of the search and enable users to explore multidimensional data using coordinated tables and graphs. Rubber band selection is used to allow the user to perform specific pattern search on the data using Euclidean distance. It is focused on multiple time series query based on examples, so preselecting an interesting pattern should be provided. That is similar to TimeClassifier [29] which requires one instance of behavior in order bootstrap the matching process. Hence, both systems require users having a general idea of what constitutes interesting patterns to specify interesting regions.

Van den Elzen et al. [27] propose a visual analytics approach for the exploration and analysis of dynamic networks by reducing graphs in time steps to points which is a similar approach to the one we take here. Burch et al. [7] stated that reducing snapshots to points was good for scalability, but the actual graph structures were not visible anymore. Lin et al. [21] mention that the method could display an overview of the evolution, but could not give reasons why they are similar or different.

Interaction techniques have been focused to enhance the time-series graph for large data. Kincaid et al. [16] apply a pixel-based display to multiple time-series graphs. It provides a compact overview by encoding the y-dimension of individual line graphs with color instead of space and displays picked graphs in detail using standard techniques. It depends on focus and context techniques. Hao et al. [9] also present a space-filling, multi-resolution matrix representation of time series, where the display is extended to two-dimensional space using the x and y-axes.

In addition, lensing techniques [15], [33] are utilized allowing to visualize the data based on the underlying user-defined regions. To allow focusing on points of interest while maintaining the context with the remaining series, time-axis is distorted to enhance segments of interest. Javed et al. [13] present stack zoom based on layout techniques. Walker et al. [28], based on Stackzoom and ChronoLens, present TimeNotes which supports interactively selection, exploration, hierarchical navigation, and comparison of time-series data. Then, they are labeled as instances of specific behaviors.

PCA, as a feature extraction method, is effectively applied to time series data [18], [26], [31], [32]. It is often utilized to reduce the dimensions of a d-dimensional dataset by projecting it onto a w-dimensional subspace where w is less than d.

In this paper, we incorporate visualization, matching, and human interaction into one system to explore, analyze and understand a large time-series data by reducing them to points, so that the user can discover similar patterns by changing window width and overlaps between windows. Through interaction, the user creates a labeling of similar patterns in the large time-series. Each pattern has its own identifying color. It does not require labelled or pre-classified data in order to start the matching process. It takes the advantages of both visualization and interaction techniques to provide a better understanding of large time series data, allowing the user to visualize and explore the big picture of the phenomenon under consideration.

III. Methodology

Our visual analytics pipeline consists of three essential stages which are **preprocessing, feature extraction and projection, and visual exploration** see Figure 3. Every step will be illustrated in detail in the following subsections. To enable analysis and exploration of time series data, we apply the sliding window approach on raw time series data. Then, the feature extraction technique is applied to project the dataset to two dimensions. The interface is designed to support analyzing, exploring and matching large time series data with linked juxtaposed views and many options that can be adjusted to assist a user to reach the desired target.

A. Preprocessing

The time series data is prepared for analysis. The sliding window approach and the constructed matrix that are resulting from it are simultaneously crucial in our process. Each row in the constructed matrix is considered as a point in high-dimensional space, and each such point in that space represents the phenomenon under consideration at a different time-interval.

1) Sliding window approach: Given a continuous time series data Q, the sliding window technique running along the time axis depends on two significant parameters which are **window size W** and **stride (offset) S**. Sliding Window is also called brute force or one-pass algorithm [23] and has been used in many time series works for instance [11], [12], [14], [27]. It is an appropriate way to deal with temporal data because it sequentially processes the raw data keeping into account its temporal behavior. This specific discretization process is largely determined by the choice of the window length and the stride between the existing window and the following window. Both parameters have their default values in our interface, and can be modified by the user. Using this approach divides the data stream into blocks, and is considered to be a fast segmentation method, where no false dismissal can happen because of the overlapping between windows.

The result of sliding window segmentation is to construct a new matrix which will be used in the next step. The process is begun by determining the left boundary of the first window (usually the first data point of a time series), which is the starting point for the window. It slides (to the right) along the time series depending on the window size (the stopping point will be the end of this window). The first row in the newly constructed matrix will be this first window. Based on the stride, the second window starting point begins, which represents the second row in the matrix. In this manner, the process is repeated until the end of series see Figure 3 (a).

Q is a time series of size n with Q = (q_1, q_2, . . . , q_i, . . . , q_n). W denotes the window size. S denotes stride (offset). We define a matrix X of the sliding windows:

$$X_{m\times n} = \begin{bmatrix} W0 \\ W1 \\ W2 \\ W3 \\ .... \\ .... \\ .... \\ Wk \end{bmatrix} = \begin{pmatrix} q_{r0}, q_{r0+1}, q_{r0+2}, \cdots\cdots, q_{r0+w-1} \\ q_{r1+1}, q_{r1+2}, q_{r1+3}, \cdots\cdots, q_{r1+w} \\ q_{r2+1}, q_{r2+2}, q_{r2+3}, \cdots\cdots, q_{r2+w} \\ q_{r3+1}, q_{r3+2}, q_{r3+3}, \cdots\cdots, q_{r3+w} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ q_{rk+1}, q_{rk+2}, q_{rk+3}, \cdots\cdots, q_{rk+w} \end{pmatrix}$$

Where $r_i = i \times S$ and $i \times S \leq n - W$

Appropriate values for the window-length and stride parameters are set interactively. The overlapping between windows is beneficial to avoid missing any data and facilitate the smooth transition between time-steps after projecting the data to the new space.

B. Features extraction and projection

The segments of time series data are treated as points in high-dimensional space (WD space). The feature-based technique is used, to reduce the feature space to lower dimensional subspaces for visualization, understanding, and analysis. Data in the lower dimensional subspace are approximated to the geometric attributes of the data in the original high-dimensional space.

Several feature based techniques have been proposed to represent features with low dimensionality for time series data. Principal Component Analysis (PCA), as an eigenvalue method, is a technique which transforms the original time series data into low dimensionality features. PCA has been utilized in visual analytics finding relationships between variables in the data, visualizing and interpreting data, and data dimensionality reduction [2], [3], [19], [22], [27].

PCA transforms data to a new set of variables whose elements are mutually uncorrelated, so it learns a representation of data that has lower dimensionality than the original input. PCA has been used as an effective dimensionality reduction method that eliminates the least significant information in the data. Hence, a complex dataset can be reduced to a lower dimension which helps to reveal the sometimes hidden, simplified dynamics that often underlie it.

Our goal can be achieved by reducing the feature space to two dimensions and back projecting the original data to the newly determined space. As a result, they will be represented as 2D points for visualization and interaction. In this situation, standardization is carried out on matrix $X_{m\times n}$ to create data with zero mean and unit variance. This standardization is important, particularly, if data was measured on different scales. Many machine learning algorithms require it to obtain the optimal performance.

The second step is calculating the covariance matrix which is a square and symmetric matrix. The covariance of the matrix $X_{m\times n}$ is $Cov_{n\times n}$, where every element illustrates the covariance between two features columns (x_{ix} and x_{iy} where $1 \leq i \leq n$). It can be obtained using equations 1 and 2:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{1}$$

$$Cov_{xy} = \frac{1}{n-1} \sum_{1}^{n} (x_{ix} - \bar{x_x})(x_{iy} - \bar{x_y}) \tag{2}$$

In our case, because we apply the sliding window approach, and our data is collected from sensors which generally measure data on convergent scales, we obtain a stationary mean over all of the features. Based on that, there is no need for standardization and, as such, the computation time for PCA is improved. Also, the covariance matrix is modified, equation 3, based on the stationary mean of the whole time-series data, $\bar{x}$. This accelerates PCA for large datasets (when changing window width), and allows interactive real-time analysis.

$$Cov_{xy} = \frac{1}{n-1} \sum_{1}^{n} (x_{ix} - \bar{x})(x_{iy} - \bar{x}) \tag{3}$$

The eigenvectors (e_1,e_2,...,e_d) and corresponding eigenvalues (λ_1,λ_2,...,λ_d) are computed using equation 4 where Cov denotes the covariance matrix, e denotes the eigenvector matrix, and λ denotes an eigenvalue. The eigenvectors are ordered descending by their eigenvalues. The goal is to preserve as much of the variance in the original data as possible in the new coordinate system. The first two eigenvectors are selected which always have the highest eigenvalues.

$$Cov\ e = \lambda\ e \tag{4}$$

$$Y = W^T X \tag{5}$$

Once we have determined the components (eigenvectors) that we will keep in our data, we take the transpose of the vector and multiply it using the original dataset equation 5 (where Y denotes the transformed dimensional samples in the new subspace, X denotes the original data, and W denotes the eigenvector matrix).

C. Visual Exploration

Visual data exploration usually follows three main steps: **overview first, zoom and filter, and then details-on-demand** which is called the Information Seeking Mantra [25]. This mantra insight fully summarizes the fundamental elements of interacting with presented information. First of all, the user needs to obtain **an overview** of the data to identify interesting patterns. Our approach facilitates the overview task using the connected scatter plot which presents the entire temporal data on one image (see Figure 2). Clusters, outliers, and patterns can be determined which are useful criteria to look for.

The two linked juxtaposed views are utilized to enable the exploration of the time series data (see Figure 2). The first view shows the time series graph, while the second shows the projection points after applying our approach. Each point in the connected scatter plot represents one sliding window in time series graph. Each point is connected by two lines; one links it to the previous point (inner) and the second one links it to the posterior point (outer), preserving the temporal ordering of the data.

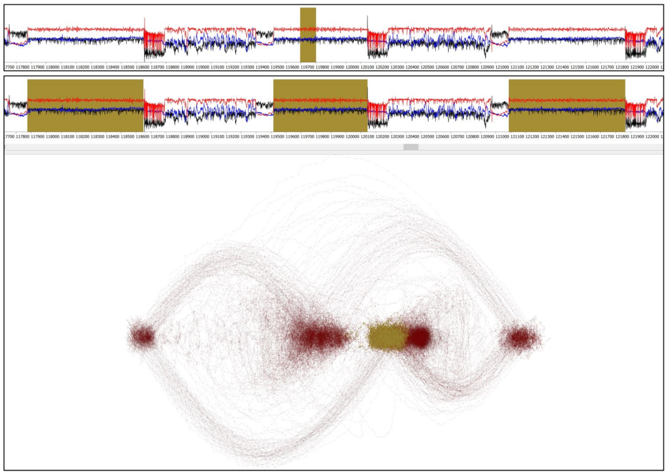


Fig. 4. Selection using similarity. The user selects part of the data from the time-series view (top). All other data that matches below a threshold based on Euclidean distance are labelled the same. In this case it corresponds to the ”Surface swimming” cluster.

Along with an overview, **the navigation tools** are supported to pan or scroll through the data. If the user wants to drill-down and access details of the data, selecting and zooming can be used to facilitate quick and interactive exploration of large datasets. Smooth zooming is used in order to explore a region of interest in both line and scatter graphs that helps user preserve their sense of position and context (see Figure 2 (c) and 2 (d)). Zooming and panning techniques are also used to support exploration, especially in scatter plot, to assist the close-up visual search of clusters.

For details-on-demand, the idea of **linking and brushing** is used to facilitate combining two different visualization techniques (time series graph and connected scatter graph). For example, the desired points can be selected and highlighted in either views which are concurrently reflected on both line and scatter graphs, hence, it allows the user to visualize, inspect, and differentiate between patterns, clusters, and outliers.

Brushed points that are highlighted in one visualization are automatically reflected in the other visualization with the same color that is selected by the user. That makes it easy to detect patterns and relationships in the large dataset. The user can apply the sequence of brush in or anywhere and in any order for example, in Figure 6, the user has brushed four areas on the connected scatter plot view. These areas are related back in the line chart demonstrating a successful ”labeling” of the data. One outlier has also been brushed (blue).

There are other options that are available such as, path extractions see Figure 2 (f) which is facilitated to track the transition intra- and inter- cluster. Sliding window size and stride can be modified.

We have also enable similarity matching using Euclidean distance (other similarity measures can be introduced). A data selection is made by the user Figure 4 (top time-series), then the Euclidean distance between the selection is made to the rest of the data. All windows below a distance threshold set by the user are considered to be matched and are given the same label Figure 4 (main display).

IV. Case Study

For the duration of the project, we collaborated with experts who provided us with large time series datasets, and provided a number of suggestions for improving and adding features. Accordingly, we implemented some options to give the user more control over exploration and analysis.

In our case study, we present the results of two time series datasets which assist to evaluate our system and demonstrate the usefulness of our technique. The first dataset is from movement ecologists to study animals behavior in their natural environment. The second dataset is the chronic obstructive pulmonary disease (COPD) data for 48 patients.

A. *Case Study 1 - Imperial Cormorant Birds*

Background Animal behavior is a rapidly growing and advancing area of study. It includes all the ways that the animals interact with other organisms and their natural environment. One of the potential ways of identifying animal behavior is through movement. Sensors, such as accelerometers, are widely used in the area of biological research to measure behavior in wild animals. The obtained data, from the attachment of tri-axial accelerometers, is analyzed to allow researchers to investigate the movement and therefore behavior of the animals.

Inspection of multiple sensors at high frequencies is time-consuming and requires a great deal of expert knowledge [4], [17], [28]–[30]. Previous work by the biologists has shown that Overall Dynamic Body Acceleration (ODBA) [8] and Vectorial Dynamic Body Acceleration (VeDBA) [5], calculated from the raw acceleration values, are good proxies for energy use. We derive the sliding window matrix for VeDBA. We also derive the sliding window matrix for the raw tri-axial accelerometer data by vectorizing them into one vector by x, y, and z order. After that, each vector is placed as a row in the constructed matrix.

Initial view We begin the analysis session by loading the raw accelerometry data which contains 173,256 data points into the visual analytics tool. The data for an Imperial cormorant exhibits five main behaviors which are Descent Phase of Dive (cluster A), Bottom Phase of Dive (cluster B), Ascent Phase of Dive (cluster C), Surface Swimming (cluster D), and Flight (cluster E). The raw accelerometer data is presented in Figure 2.a and the connected scatter plot of our approach in Figure 2.b (prior to the coloured selections). Five clusters are clearly seen and the transitions between them. Each point in the plot represents the animal movement for a particular duration. Other methods, such as k-means clustering require the number of clusters to be known in advance which is difficult to determine in large datasets. By applying our approach, this data presents clusters which exactly correspond to the five main behaviors in the dataset.

Interaction Users can select in either view to determine areas of interest, and can freely zoom into detail, see Figure 2.c and 2.d. Using the brushing tool, the biologist selects the leftmost cluster (Figure 2.c) and assigns the yellow colour. The time-series chart view updates to highlight in yellow all data associated with the cluster. Each of the remaining clusters are selected in turn, are assigned a color, and create the highlights in the time-series chart view. Through this interaction, the expert is able to confirm each of the clusters from our visualization correspond to one of the behaviors in the raw accelerometry data. This has great potential to speed up the manual labelling of time-series data.

In the time-series chart view, the rubberband selection is used to highlight a time sequence of any duration. The corresponding points that represent the sequence are highlighted in the scatter plot, which can help to identify the position within the clusters of known features in the raw data.

Edges between clusters The user can also select bundles of edges (Figure 2.f). In addition to brushing points, we allow the user to direct a region growing selection out from selected source points. Given a source point not part of one of the already highlighted clusters, a selection region is grown from the point in both temporal directions until we reach a point which is part of an existing cluster. The source selection can contain multiple points. Using this approach, the user is able to select a whole bundle of edges that are the transitional paths between two clusters.

For instance, the transitions (Figure 2.f) are selected which illustrate the dominant transition between Surface Swimming (cluster D) to Descent Phase of Dive (A). We also see dominant transitions of Descent (A) to Bottom Feeding (B), Bottom Feeding (B) to Ascent (C), and Ascent (C) to Surafce (D). Cluster E (flight) dominates the start and end of the data, but also occurs as at several shorter intervals throughout the data, therefore we see some activity between those clusters. We can interpret the well-defined edges between clusters as repetitive behavior that moves through those states with high frequency. Weaker edges indicate less frequent behavior. Both types of transitions can be labelled quickly using our interface for further analysis.

B. *Case Study 2 - Breathing Patterns*

Pulmonary fibrosis (PF) is a restrictive lung disease that can alter breathing patterns due to pathological changes in lung mechanics. Inspiratory and expiratory (flow time-series data) is taken from 48 participants, (n= 18 healthy lungs and n= 30 with PF). Each participant has around 12,000 flow readings. Looking for patterns in the time-series data is exceedingly important and complex process to reveal abnormal any tidal breathing patterns in the PF. Two problems arise in the understanding of the data. First, since each person has a long time-series, it is difficult to compare all the individual

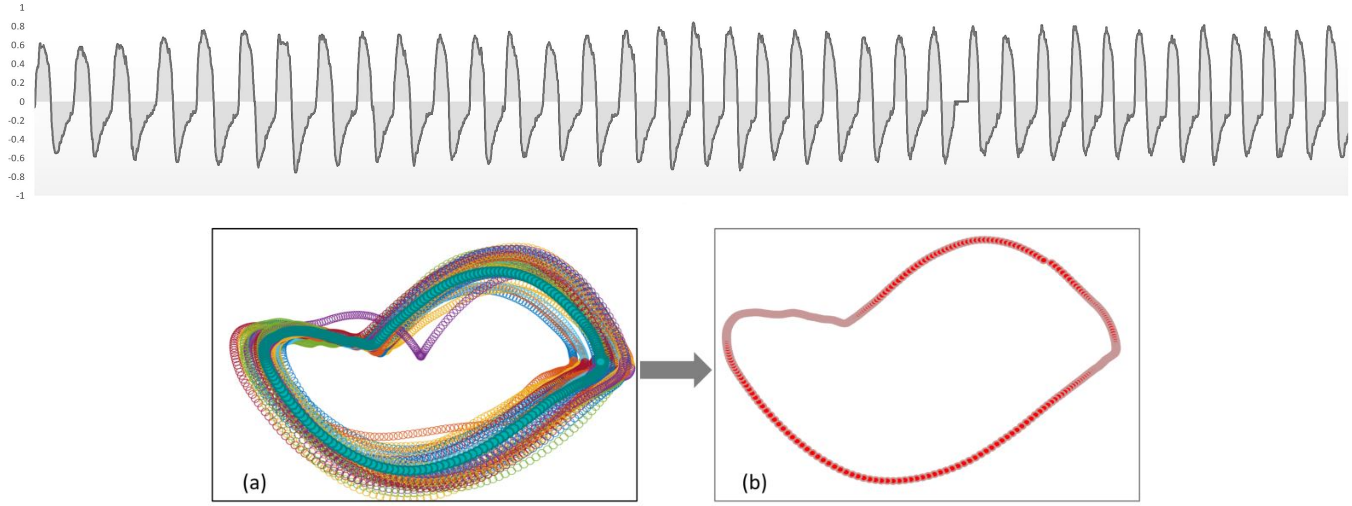


Fig. 5. Top: A portion of the time series graph for breathing which contains 11000 flow data points (inhalation and exhalation for one person). About two-thirds along the time-series view there is an interrupted breath (e.g. an expiratory pause). Bottom: (a) time series data after applying our approach, where each colored loop represents one inhalation and exhalation (the interrupted breath stands out in magenta), (b) one selected inhalation and exhalation loop. Also note, the slightly shallower orange path corresponds to another small interruption in a breath (not in the subset of data shown).

breaths against each other to look for abnormalities. Secondly, for so many participants, it is difficult to compare one person with another, or several others.

Additional to being able to compare across the time-series, another functional requirement is to be able to remove anomalous patterns easily. For example, a patient may swallow, or cough on expiration or sigh on inspiration interrupting their normal breathing pattern and rhythm. It would be useful to remove away such data so that it does not participate in any of the further statistical analysis that takes place on the data.

By applying our method to this dataset, every inspiratory and expiratory breath are represented as one loop see Figure 5.b. Finding irregular patterns is much easier using our method because all tidal breathing waveforms are presented in one view (Figure 5.a). This fulfills a requirement to see all breaths in one view. Outliers within this view corresponds to problematical breaths. By brushing series of points, it can be confirmed that the purple outlier (Figure 5.a) corresponds to an interrupted breath (visible about two-thirds along the time-series view). Also, the slightly shallower orange line corresponds to another interruption on the expiration phase of another breath. Such breaths can be removed from further analysis.

V. Conclusion

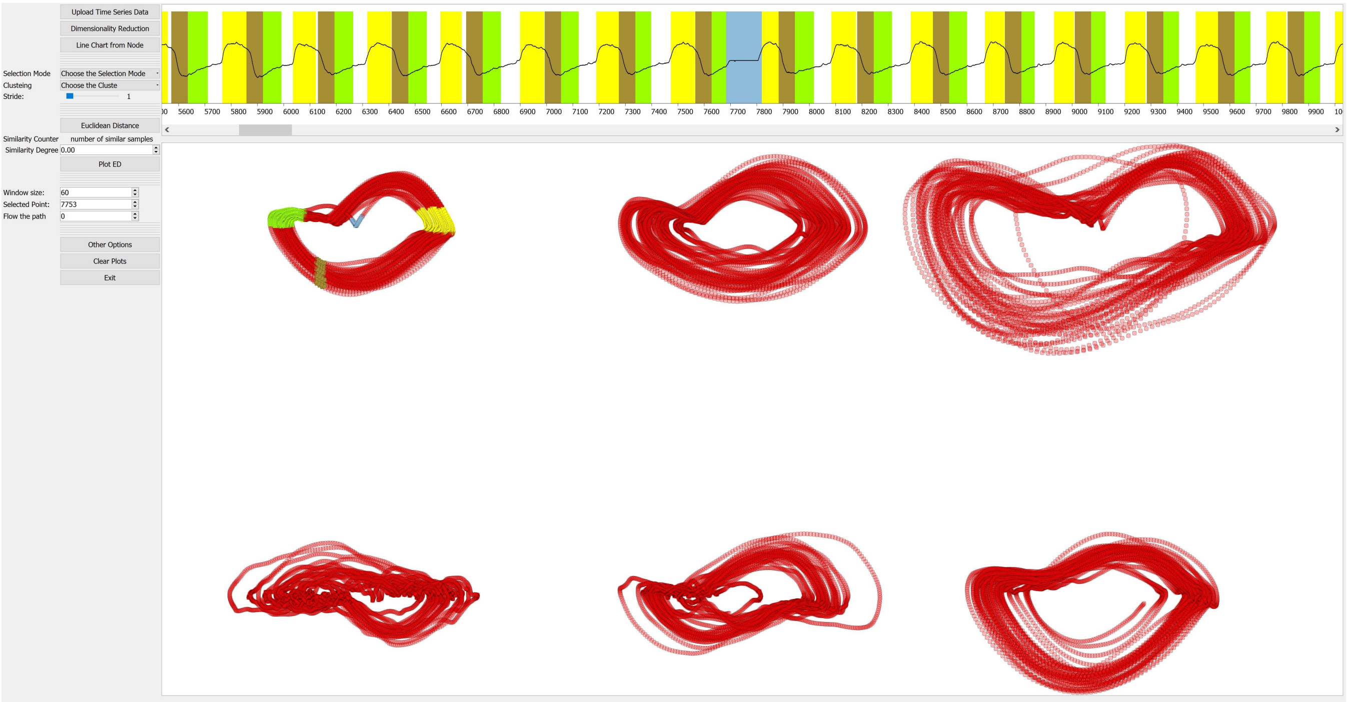
In this paper, we present a visual analytics system and approach that provides a more effective working environment for the exploration, analysis, and presentation of large time-series data. Our approach offers two primary uses.

Identifying Clusters: Dimension reduction on the sliding window approach produces scatter plots where close points have similar characteristics and naturally form clusters. The approach does not need pre-knowledge of the number of clustsers in the data like k-means approaches. The users can brush points within clusters to quickly label the data. This greatly accelerates the process of labelling behaviours within the data for domain experts. It is also easy to select bundles of transition edges between clusters, again leading to efficient labelling of similar characteristics in the raw data.

Identifying Outliers: It is difficult to detect outliers in long time-series data, as it is requires being able to relate different parts of the data together. In particular, when there are many repetitive patterns, it is difficult to separate them on any detailed feature of the data. Using this approach facilitates outlier detection. The outlier paths are visually detectable. In Figure 7, there is a clear outlier in the breathing pattern for this subject. The user brushes the scatterplot view (yellow), which highlights the source in the time-series view. It is clear this breath has some artifact compared to the others. In this case, the affected data can be labelled to be excluded from further statistical processing since it was certainly due to a sigh.

Our results using the tool with expert users indicate that it is a promising way to handle large temporal datasets. We have shown the effectiveness of our work by applying it to two different large time series datasets demonstrating the ability to relate features separated by large time-dimension.

For future work, we plan to enhance our system in several ways. Currently, we use PCA as a technique for dimensionality reduction, therefore, we plan to use different dimensionality reduction techniques such as t-SNE. Also, we wish to apply clustering algorithms e.g., hierarchical algorithms on the data after applying our approach.



Selection Mode: Choose the Selection Mode
Clusteing: Choose the Cluste
Stride: 1
Similarity Counter: number of similar samples
Similarity Degree: 0.00
Window size: 60
Selected Point: 7753
Flow the path: 0

Fig. 6. The main user interface of our system on data from a deployment on pulmonary fibrosis data. Top: zoomed time-series graph with overlaid colored regions corresponding to the chosen clusters in scatter plot. The user can zoom in both graphs to see more detail. The left side of the interface shows some buttons and parameters which can be modified based on the user desires. Each cluster, outlier, and transition can be colored with a different color to be identified and compared.

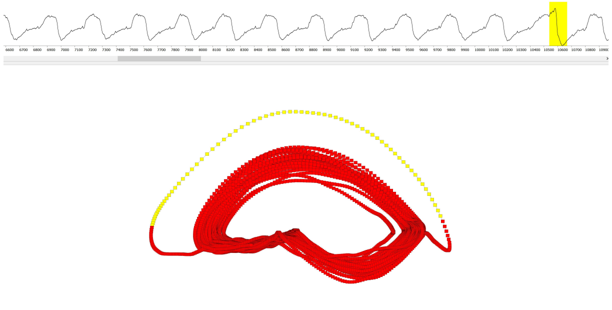


Fig. 7. Yellow highlight in the time series graph and yellow points in the scatter plot indicate that the breathing at that moment was totally different than the rest of the tidal breathing which is obviously clear in time series graph.

References

[1] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011.

[2] Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE transactions on visualization and computer graphics*, 19(12):2257–2266, 2013.

[3] Jürgen Bernard, Nils Wilhelm, Maximilian Scherer, Thorsten May, and Tobias Schreck. Timeseriespaths: Projection-based explorative analysis of multivariate time series data. 2012.

[4] Owen R Bidder, Hamish A Campbell, Agustina Gómez-Laich, Patricia Urgé, James Walker, Yuzhi Cai, Lianli Gao, Flavio Quintana, and Rory P Wilson. Love thy neighbour: automatic animal behavioural classification of acceleration data using the k-nearest neighbour algorithm. *PloS one*, 9(2):e88609, 2014.

[5] Owen R Bidder, Lama A Qasem, and Rory P Wilson. On higher ground: How well can dynamic body acceleration determine speed in variable terrain? *PLoS One*, 7(11):e50556, 2012.

[6] Paolo Buono, Aleks Aris, Catherine Plaisant, Amir Khella, and Ben Shneiderman. Interactive pattern search in time series. In *Visualization and Data Analysis 2005*, volume 5669, pages 175–187. International Society for Optics and Photonics, 2005.

[7] Michael Burch, Marcel Hlawatsch, and Daniel Weiskopf. Visualizing a sequence of a thousand graphs (or even more). In *Computer Graphics Forum*, volume 36, pages 261–271. Wiley Online Library, 2017.

[8] Adrian C Gleiss, Rory P Wilson, and Emily LC Shepard. Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure. *Methods in Ecology and Evolution*, 2(1):23–33, 2011.

[9] Ming C Hao, Umeshwar Dayal, Daniel A Keim, and Tobias Schreck. Multi-resolution techniques for visual exploration of large time-series data. In *EUROVIS 2007*, pages 27–34, 2007.

[10] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[11] Hesam Izakian and Witold Pedrycz. Anomaly detection and characterization in spatial time series data: A cluster-centric approach. *IEEE Transactions on Fuzzy Systems*, 22(6):1612–1624, 2014.

[12] Dominik Jäckle, Fabian Fischer, Tobias Schreck, and Daniel A Keim. Temporal mds plots for analysis of multivariate data. *IEEE transactions on visualization and computer graphics*, 22(1):141–150, 2016.

[13] Waqas Javed and Niklas Elmqvist. Stack zooming for multi-focus interaction in time-series data visualization. In *Visualization Symposium (PacificVis), 2010 IEEE Pacific*, pages 33–40. IEEE, 2010.

[14] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296. IEEE, 2001.

[15] Robert Kincaid. Signallens: Focus+ context applied to electronic time

series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):900–907, 2010.

[16] Robert Kincaid and Heidi Lam. Line graph explorer: scalable display of line graphs using focus+ context. In *Proceedings of the working conference on Advanced visual interfaces*, pages 404–411. ACM, 2006.

[17] Agustina Gómez Laich, Rory P Wilson, Flavio Quintana, and Emily LC Shepard. Identification of imperial cormorant phalacrocorax atriceps behaviour using accelerometers. *Endangered species research*, 10:29–37, 2008.

[18] Ragnar H Lesch, Yannick Caillé, and David Lowe. Component analysis in financial time series. In *Computational Intelligence for Financial Engineering, 1999.(CIFEr) Proceedings of the IEEE/IAFE 1999 Conference on*, pages 183–190. IEEE, 1999.

[19] Jie Li, Kang Zhang, and Zhao-Peng Meng. Vismate: Interactive visual analysis of station-based observation data on climate changes. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 133–142. IEEE, 2014.

[20] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Jeffrey P Lankford, and Daonna M Nystrom. Viztree: a tool for visually mining and monitoring massive time series databases. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1269–1272. VLDB Endowment, 2004.

[21] Lijing Lin, Siming Chen, Fan Hong, Chufan Lai, Shuai Chen, and Xiaoru Yuan. Graphlda: Latent dirichlet allocation-based visual exploration of dynamic graphs.

[22] Tao Lin, Fangzhou Guo, Yingcai Wu, Biao Zhu, Fan Zhang, Huamin Qu, and Wei Chen. Tievis: Visual analytics of evolution of interpersonal ties. In *International Conference on Technologies for E-Learning and Digital Entertainment*, pages 412–424. Springer, 2016.

[23] Miodrag Lovrić, Marina Milanović, and Milan Stamenković. Algoritmic methods for segmentation of time series: An overview. *Journal of Contemporary Economic and Business Issues*, 1(1):31–53, 2014.

[24] Patricia Ordóñez et al. Visualizing multivariate time series data to detect specific medical conditions. In *AMIA Annual Symposium Proceedings*, volume 2008, page 530. American Medical Informatics Association, 2008.

[25] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The Craft of Information Visualization*, pages 364–371. Elsevier, 2003.

[26] Ashish Singhal and Dale E Seborg. Clustering multivariate time-series data. *Journal of chemometrics*, 19(8):427–438, 2005.

[27] Stef van den Elzen, Danny Holten, Jorik Blaas, and Jarke J van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE transactions on visualization and computer graphics*, 22(1):1–10, 2016.

[28] James Walker, Rita Borgo, and Mark W Jones. Timenotes: a study on effective chart visualization and interaction techniques for time-series data. *IEEE transactions on visualization and computer graphics*, 22(1):549–558, 2016.

[29] James S Walker, Mark W Jones, Robert S Laramee, Owen R Bidder, Hannah J Williams, Rebecca Scott, Emily LC Shepard, and Rory P Wilson. Timeclassifier: a visual analytic system for the classification of multi-dimensional time series data. *The Visual Computer*, 31(6-8):1067–1078, 2015.

[30] Rory P Wilson, Craig R White, Flavio Quintana, Lewis G Halsey, Nikolai Liebsch, Graham R Martin, and Patrick J Butler. Moving towards acceleration for estimates of activity-specific metabolic rate in free-living animals: the case of the cormorant. *Journal of Animal Ecology*, 75(5):1081–1090, 2006.

[31] Kiyoung Yang and Cyrus Shahabi. A pca-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 65–74. ACM, 2004.

[32] Kiyoung Yang and Cyrus Shahabi. On the stationarity of multivariate time series for correlation-based data analysis. In *Data Mining, Fifth IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[33] Jian Zhao, Fanny Chevalier, Emmanuel Pietriga, and Ravin Balakrishnan. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011.