Clustering and Classification for Time Series Data in Visual Analytics: A Survey

MOHAMMED ALI[1,2], **ALI ALQAHTANI**[1,2], **MARK W. JONES**[1], **AND XIANGHUA XIE**[1], (Senior Member, IEEE)
[1]Department of Computer Science, Swansea University, Swansea SA1 8EN, U.K.
[2]Department of Computer Science, King Khalid University, Abha 61421, Saudi Arabia

Corresponding author: Mohammed Ali (884715@swansea.ac.uk)

ABSTRACT Visual analytics for time series data has received a considerable amount of attention. Different approaches have been developed to understand the characteristics of the data and obtain meaningful statistics in order to explore the underlying processes, identify and estimate trends, make decisions and predict the future. The machine learning and visualization areas share a focus on extracting information from data. In this paper, we consider not only automatic methods but also interactive exploration. The ability to embed efficient machine learning techniques (clustering and classification) in interactive visualization systems is highly desirable in order to gain the most from both humans and computers. We present a literature review of some of the most important publications in the field and classify over 60 published papers from six different perspectives. This review intends to clarify the major concepts with which clustering or classification algorithms are used in visual analytics for time series data and provide a valuable guide for both new researchers and experts in the emerging field of integrating machine learning techniques into visual analytics.

INDEX TERMS Time series data, clustering, classification, visualization, visual analytics.

I. INTRODUCTION AND MOTIVATION

Recent years have seen an increasing use of time-oriented data in many fields such as networks and systems, meteorology, social media, behavior analysis, trajectory data, biological science, finance, and the like, where data is measured at a regular interval of (real) time. In this research work, we focus on time series data; it is therefore important to agree on a formal definition. Time series data is defined as an ordered collection of observations or sequence of data points made through time at often uniform time intervals [1]. Also, because of its diversity of sources, its complexity, and its various underlying structures, we categorize time series data, used in our surveyed papers, into four categories based on their structure: univariate, multivariate, tensor fields and multifield.

Machine learning gives computers the ability to learn without explicit programming [2]. Alpaydin [3] gives a concise description of machine learning, which is "optimizing a performance criterion using example data and past experience".

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.

Data plays a major role in machine learning where the learning algorithm is utilized to discover and learn knowledge or properties from the data (learn from experience) without depending on a predetermined equation as a model [4]. In **supervised** learning, the dataset (the training set) is composed of pairs of input and desired output and learning aims to generate a function that maps inputs to outputs. Each example is associated with a label or target. In **unsupervised** learning, the dataset (the training set) is composed of unlabeled inputs without any assigned desired output and the aim is to find hidden patterns or substantial structures in data [5]. There are different types of supervised and unsupervised machine learning techniques and under each approach has different algorithms taking various approaches to learning. Our focus in this work will be on classification as a supervised learning technique and clustering as an unsupervised learning technique with time series data.

Sacha *et al.* [6] highlight two main functions for machine learning. The first is to transform unstructured data into a form which facilitates human exploration, analysis and understanding. The second is to utilize unsupervised or semi-supervised algorithms to direct the analysis itself by

recommending the best visualizations, verification, successions of steps in the exploration, etc., where the algorithm can automatically discover complex patterns from the raw data directly. This user-centric approach of interactive visualization utilizes human vision scalability for analyzing, exploring and understanding such data. It also assists data analysts in solving complex problems interactively by integrating automated data analysis and mining, such as machine learning-based methods, with interactive visualizations [7].

Machine learning algorithms provide a collection of automated analyses which can be much more efficient, accurate and objective in solving time series tasks. Machine learning also focuses on prediction [8] which has useful and widespread real-world applications.

The machine learning and visualization communities have been addressing time series issues from different perspectives. Machine learning has a strong algorithmic focus while interactive visualization has a strong human/visualization focus [9]. Therefore, the essential difference between the fields, is the role of the user in data exploration and modeling. In machine learning, the goal is to dispose of the user, so everything is automated. In this case, the user can play a limited role such as selecting the type of algorithm, where their influence should be restricted to a minimum. In an interactive visualization, a completely opposite point of view is offered, where visual representations are leveraged by the user to extract knowledge from the data, discover patterns, adjust models of the data under user steering. This main difference in philosophy may explain why both communities have remained relatively disconnected [10].

Based on the above, there is a strong incentive for both communities to be synergized in order to make progress and benefit from one another [7]. Combining automated analysis methods and interactive visualization has been shown to be an efficient approach for visual analytics. The visual analytics process aims to tightly couple automatic analysis methods and interactive visualization in order to gain knowledge from raw data and present a chance for analysts, through interaction tasks, to analyze, explore, reason, discover, and understand the data.

A. SURVEY SCOPE AND INTENDED AUDIENCE
Our focus will be on two important machine learning tasks, namely clustering and classification, and how they are integrated into visual analytics systems for time series data. From a broader point of view, existing works come from two different fields which can be classified into two categories: data mining approaches [1], [11]–[14] and visualization approaches [15]–[17].

1) FROM A DATA MINING PERSPECTIVE
Several surveys are available on clustering and classification for time series data. Liao [11] and Aghabozorgi *et al.* [12] provide an overview on clustering time series data. Xing *et al.* [13] present a review for time series data classification. Moreover, Yahyaoui and Al-Mutairi [14] also

discuss some classification algorithms that are used with sequence data. Fu [1] provides an overall picture of the current time series data mining techniques including clustering and classification tasks. These previous works discuss in detail a wide range of clustering and classification algorithms that have been proposed and employed on time series data with a strong algorithmic focus. However, user influence is not considered in most of these works.

2) FROM THE VISUALIZATION PERSPECTIVE
Aigner *et al.* [15] provide a complete classification scheme for time-oriented data. A large part of their work involves a structured survey of existing techniques for visualizing time-oriented data, illustrated with numerous examples. Bach *et al.* [16] review a range of temporal data visualization techniques and classify them from a new perspective by depicting each technique as series of operations performed on a conceptual space-time cube. However, their work does not provide much guidance for interaction design. Additionally, Ko *et al.* [17] present a survey that categorizes financial systems from the visual analysis perspective. Their focus is on financial data, which is one of several different kinds of time series data. In contrast, our work looks at time series data in general, primarily emphasizing clustering and classification tasks with a variety of visual analytics systems, which focus on combining machine learning algorithms and visualization techniques.

3) TOWARDS INTEGRATION AND CONVERGENCE
The idea of integration between machine learning algorithms and interactive visualization has been encouraged and promoted from both the visualization and machine learning communities. For example, several recent initiatives have been put into place to bring the two domains closer, such as the annual CD-MAKE conference and the MAKE-Journal [18], [19]. The recently organized Dagstuhl Seminars titled ''Information Visualization, Visual Data Mining and Machine Learning'' (12081) [10] and ''Bridging Information Visualization with Machine Learning'' (15101) [7] are other examples of efforts to bring researchers from both domains together to discuss important challenges and corresponding solutions for integrating the two fields.

To understand this interplay between both domains, the working group in the Dagstuhl Seminar ''Bridging Information Visualization with Machine Learning'' (15101) [7] developed a framework which conceptualizes how the incorporation of interactive visualizations and machine learning algorithms can be performed. This framework was inspired by Keim *et al.* [20] visual analytics framework. The group attempts to identify aspects of machine learning by the user such as adjusting the parameters of models or switching between different model kinds. Montes *et al.* [21] present a work which is considered as one of the groundbreaking works in this trend. They combine visualization with machine learning techniques (clustering and classification) over time series data to understand the behavior of complex distributed
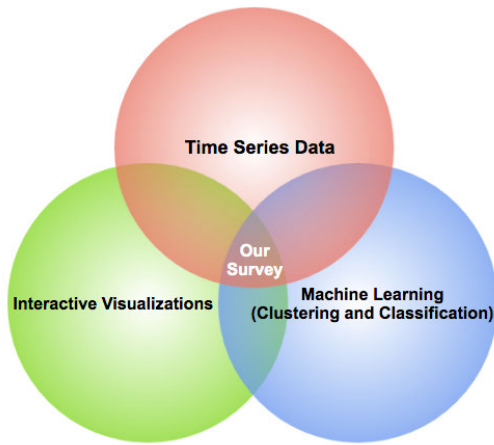
FIGURE 1. This survey focuses on the intersection between time series data, machine learning techniques (clustering and classification), and interactive visualizations.

systems. Recently, Sacha *et al.* [22] developed an ontology which maps out all major processes in machine learning and aims to provide visual analytics practitioners with a means to ''navigate'' the intricate landscape of machine learning, in order to uncover aspects which might be improved by introducing more machine or human capabilities.

To the best of our knowledge, there are no previous survey papers that offer a systematic review of the literature for time series clustering and classification that combine visualization techniques and machine learning algorithms for visual analytics. In this work, we specifically look at the convergence between automatic methods and interactive exploration, and how such automatic methods have been used in visual analytics systems (as shown in Fig. 1).

We provide a comprehensive and detailed survey on clustering and classification in visual analytics systems that have been applied to time series data. Although a large enough body of literature has covered the clustering and classification of time series data, their focus is either on algorithms or interactive visualization. However, the idea of integration and convergence between both domains is beneficial; for instance, clustering is one of the most popular algorithms to have been incorporated into visual analytics systems. Since visual representations are quite significant for interpreting and understanding the characteristics of clusters output by algorithms, direct adjustment of clustering algorithms is often facilitated through interactive interfaces that present new results ''on-demand'' [23].

4) CLUSTERING AND CLASSIFICATION OF TEMPORAL AND NON-TEMPORAL DATA

For time series data the presence of noise, its high dimensionality and high feature correlation pose challenges for designing effective and efficient clustering and classification algorithms compared to data without a temporal component [15], [24].

Analyzing time series data is nontrivial and can even vary over time due to complex interrelations between time series

variables. Xing *et al.* [13] mention three major challenges for time series analysis especially in classification. First, many methods can only take input data as a vector of features. Unfortunately, there are no explicit features in sequence data. Second, feature selection is not easy because the dimensionality of the feature space can be high and computation can be costly. Third, since there are no explicit features, building an interpretable sequence classifier is burdensome in some applications.

Computing the similarity between two data objects is considered one of the main differences between clustering and classification of temporal and non-temporal data [11], [25]. The unique characteristics of time series data such as noise, including outliers and shifts and the varying length of time series has made similarity measures one of the main challenges for clustering and classification of time series data [12]. When dealing with time series data, the biggest challenge lies in replacing the distance/similarity measure for static data with a suitable one for time series data because it may be scaled and translated differently both on the temporal and behavioral dimensions [24], [26]. In the context of visualization, classification and clustering tasks share a common goal which is data abstraction. This is for subsequent visualization, to decrease the workload when computing visual representations and to minimize the perceptual effort required to interpret them.

Keim *et al.* [27] present the visual analytics mantra: "Analyse First - Show the Important - Zoom, Filter and Analyse Further - Details on Demand". Accordingly, it is not enough to only recover and display the data using visualization techniques; rather, it is essential to analyze the data according to its value of interest, displaying the most relevant aspects of the data, and at the same time providing interaction techniques, which assist the user to gain details of the data on demand. Automatic analysis techniques are critical to the visual analytics process and are essential to incorporate in parallel with the interactive visual representation. Also, analysis techniques such as feature selection, dimensionality reduction and clustering, support gaining insight into data and support human cognition to process large volumes of data, enabling visualization to scale. Visual analytics also allows users to interact with these algorithms, in some cases, through interactive interfaces such as directing the modification of algorithms, accepting user input or switching between algorithms and display new results ''on-demand'' [20].

5) CLASSIFICATION

We classify the surveyed papers from six different perspectives, these being Time series Data Structures, Similarity Measures and Feature Extraction for Time series Data, Time series Analysis Techniques (Clustering and Classification), Visualization Analysis, and Evaluation Approaches.

6) SURVEY SCOPE

A variety of concepts and methods are involved in achieving the goal of extracting useful structures from large volumes of

TABLE 1. The selected visual analytics papers of time series data. The table provides an overview of the surveyed papers regarding similarity measures and feature extraction, time series analysis techniques (clustering or classification), visualization techniques, visualization tasks and interaction methods, evaluation approaches, and distribution of papers by year of publication.

The table presents a matrix of surveyed papers (columns, spanning years 2006–2018) against the following categories and row labels:

- **F&S:** Feature Based; Similarity Based
- **Analysis Techniques**
  - Clustering: Partition; Hierarchical; Model Based; Density Based; Grid Based; Other
  - Classification: Decision Tree; SVM; NN; Other
- **Visualization Techniques**
  - Bar: Bar Graph; Ripple Graph; Histogram
  - Area: Stacked Graph; River Graph; Stream Graph
  - Circle: Pie Chart; Radial Chart
  - Line: Line Plot; Parallel-Coordinate; Time-of-Saliency; KnotLines
  - Geographic Maps
  - Grid, and Matrix: Heat Map; Ranking View; Calendar map; Space-Time Cube; Tessellation
  - Point: Pixel; Bubble chart; Scatterplots
  - Trees and Network: Dendrogram; Link Nodes
  - Glyphs
- **Visualization Tasks and Interaction Methods**
  - Analysis: Consume; Produce
  - Search: Lookup; Browse; Locate; Explore
  - Query: Identify; Compare; Summarize
  - Encode: Arrange; Map
  - Manipulate: Select; Navigate; Change; Filter; Aggregate
  - Introduce
- **Evaluation:** Case Study; Usability Study; Controlled User Study; Algorithms Performance; Heuristic

data including statistics, machine learning, neural networks, data visualization, pattern recognition, and high-performance computing [15]. Time series analysis is dominated by traditional statistical methods such as autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) as well as machine learning techniques such as k-means and support vector machine (SVM). Machine learning methods have also shown ability for time series analysis. They also enable analysis tasks such as clustering, classification and prediction [28], [29]. Recently, Neural networks have been increasingly used with sequential data such as text data analysis where the recurrent neural network (RNN) has received popularity.

Aigner *et al.* [15] gave a brief overview of analytical methods for time-oriented data including clustering, classification, search and retrieval, pattern discovery and prediction where visualization of temporal data can highly benefit from the analytical support. In this survey, we focus on clustering and classification. Other analytical tasks such as search and retrieval and pattern discovery are indirectly addressed by our inclusion of similarity measures, clustering and classification since these operations are the bases of pattern discovery or search. Other analysis tasks that are not in the focus in this survey, but are widely used in the context of time

series analysis, include prediction which targets to deduce from data collected in the past and display how the data will develop in the future. Linear regression, recurrent neural network (RNN), and Long short-term memory (LSTM) are the most recently used techniques for this task besides the common statistical techniques such as the autoregressive moving average model (ARIMA) and box-Jenkins method.

To fulfill the scope of our survey, we have selected papers which focus on machine learning algorithms for time series clustering and classification tasks in visual analytics systems. The sixty-five publications which have been selected span a period of thirteen years. For all papers, we pay attention to time series similarity measures and feature extraction, clustering and classification algorithms, and visual analytics. We categorize the nature of time series data and evaluation techniques. Our findings on these are summarized in Table 1. Papers that focus on time series text visualization are out of our survey's scope.

B. INTENDED AUDIENCE
The intended audience of this survey are those who already have a background in visualization and possibly want to know more about machine learning tasks, in particular clustering and classification. These tasks could help them

FIGURE 2. The time series data pipeline used to structure our surveyed papers. From the surveyed papers the items in the cloud are usually integrated into one visual analytics system which is evaluated using the various evaluation approaches surveyed in Section VI.

analyze, understand and visualize time series data. As a result, we do not go into detail about visualization (visualization techniques or visualization tasks and interaction methods) but focus more on machine learning tasks (clustering and classification) and how these algorithms have been adapted into visual analytics systems.

C. SEARCHED VENUES
For paper collection, we mainly used IEEE Xplore (e.g. TVCG, VAST and PacificVis), Springer (e.g. Visual Computer), ACM, Wiley (which includes Eurovis papers), ScienceDirect and SAGE. Using IEEE Xplore, forty-three papers were obtained mostly from IEEE Transactions on Visualization and Computer Graphics, IEEE on Visual Analytics Science and Technology (VAST), and IEEE Pacific Visualization Symposium (PacificVis). We include six papers from Springer, and six papers from ACM. The other eleven examined papers have been obtained from other digital libraries.

D. SURVEY STRUCTURE
Figure 2 shows the structure of this survey, which is derived from the main steps of the selected papers. We start with time series data structures where we provide a general classification for time series data. All data structures, as described in Section II, refer to the main definition of time series data, and this section answers questions such as how time series data structures are different, along with providing some examples of this kind of data.

In Section III, we discuss similarity measures and feature extraction which are important for time series data as, usually, the quality of analysis techniques (clustering and classification) are significantly influenced by its selection. Moreover, in this section we show how these techniques, along with clustering and classification techniques, have been adapted to gain and visualize relevant knowledge from the data.

Section IV reviews the time series analysis tasks. We provide a comprehensive explanation for popular clustering and classification algorithms that have been used in the surveyed visual analytics papers, how they are used with time series data, and how they have been adapted to interactive visualization.

Section V summarizes visualization techniques, visualization tasks and interaction methods that are used in surveyed visual analytics systems. Some of these techniques and tasks are beneficial for time series data, while others are shared when working with other kinds of data. We focus more on illustrating how these techniques and tasks are performed and adapted to assist in analyzing time series data. The evaluation approaches for the surveyed visual analytics systems are discussed in Section VI.

Our survey presents a structured review of the concept of integrating interactive visualizations and analysis techniques (clustering and classification) into the visual analytics systems for time series data. Through this, we have determined different research trends as well as some of the limitations and challenges involved in the integration and convergence of machine learning algorithms and interactive visualization. These are summarized in Section VII.

II. TIME SERIES DATA STRUCTURES
We classify time series data that has been used in our surveyed papers into four categories. This classification can be

subsumed under the concepts of univariate, multivariate, tensor fields, and multifields. Hotz and Peikert [30] discuss the complex structure of scientific data and provide a clear definition of a multifield. Our four types or categories are generalized to include many related subtypes of time series data structures in order to achieve a comprehensive classification for time series data structures that can be embodied in visual analytics systems. The prevalent representatives in our surveyed papers are multivariate time series and tensor fields.

A. UNIVARIATE

The univariate time series is a sequence that contains only one data value per temporal primitive [15], [31]. It is a field of a single variable captured or observed through time. Temperature in a city spanning a period of time is a clear example of this type of data structure.

B. MULTIVARIATE

Multivariate time series is a set of time series which have the same timestamps [15], [31]. This kind of time series data structure is an array of variables or numbers at each point in time and can be a collection of multiple univariate captured through time, such as temperature and pressure readings, or associative multivariate, such as 3-D acceleration measured from a tri-axial accelerometer, where each component of the multivariate has the same units and sensor source. As time series data structures are an ordered collection of observations or sequence of data points made through time, most of the surveyed papers adopt this type of structure. This special type of multivariate time series data is relevant in many application fields including biology, medicine, finance and animation. Multivariate time series data have been also used in manufacturing systems and predictive maintenance [32], [33]. In the surveyed visual analytics papers, time series data, e.g., obtained from gene expression measurement [34]–[37] can be used by biologists to understand the correlation between different types of genes, analyze gene interactions, and compare regulatory behaviors for interesting genes. Moreover, medical experts utilize time series data e.g., blood pressure measurements [38], to understand and deal with different cases such as monitoring illness progression, and understanding ecological and behavioral processes related to a disease which may lead to improved disease diagnoses. Furthermore, time series data, e.g., obtained from sampled transactions over a period of time [39]–[41], stock markets [42], [43], and international financial markets [44], [45] can be used in the financial field and is usually analyzed to understand and forecast the market situation. It is useful to find correlations between the data and test hypotheses about the market, which helps to make the best decisions at the appropriate time under different business and economic circumstances. A multivariate can also present time series data obtained from various data sets including metadata e.g. patient records [46], [47], employment records [48], [49], and others [50], [51].

C. TENSOR FIELDS

These are an array of data arranged on a regular grid with a variable number of axes [4]. They can be described as a quantity which is associated to each point in space-time as it has been extended to functions or distributions linked to points in space-time [30]. Dealing with spatio-temporal data, this type of time series data structure is generalized to include many related subtypes: time series of graphs and networks, time series of spatial positions of moving objects, and time series of spatial configurations/distributions.

1) TIME SERIES OF GRAPH AND NETWORK

Time series data in the form of networks consist of associated attributes such as nodes and edges that reflect different kinds of behavior over time. Node or edge attributes of dynamic graphs can be introduced as time series. This kind of time series data helps understand different temporal patterns and evaluate the network dynamics in general [52]–[56]. The network view helps to visualize the connectivity of the sensors, which can enhance analysis, detection and exploration. As each machine (e.g. engines or computers) typically consists of a large number of sensors that produce massive data, time series data can be obtained from the nodes of such machines over a period of time, such as CPU load, memory usage, network load, and data center chiller sensor, helping to improve the understanding of how machines are used in practice and analyze the performance and behaviors of such systems [57]–[63]. Indeed, analyzing this data helps users and experts understand and evaluate the network dynamics.

2) TIME SERIES OF SPATIAL POSITIONS OF MOVING OBJECTS

Spatial positions of moving objects data with an associated time component classifies as trajectory data. It presents different places over time, providing a clear idea of spatio-temporal changes. A combination of interactive visualizations and automated analysis has together been shown to be an efficient approach in analyzing, tracking, and representing this type of data in order to understand and recognize the mobility of a diversity of moving objects, such as vehicles [64]–[71], and aircraft [66], [67], which can lead to path discovery, movement analysis, and location prediction.

3) TIME SERIES OF SPATIAL CONFIGURATIONS AND DISTRIBUTIONS

Being able to extract useful insight from time series of spatial distributions and configurations is becoming more important due of the massive growth in data science and the rapid advancement of many technologies. In our surveyed papers, we consider discovering behavioral patterns and finding interesting events that might take place in certain municipalities [72] and public or business sectors as spatial configurations and distributions. This identification of regular configurations and distributions over time is represented by a total number of events and behaviors extracted from

a chosen spatial scale. Personal mobility behaviors and movement patterns [73]–[81], behaviors of animals [82], [83], pattern changes in climate (weather) and the ozone layer [81], [84]–[90], and behavior capture data made through time at often uniform time intervals [91]–[96] can be regarded as instances for this type of data structure that take a place in specific spatial identification.

D. MULTIFIELD
This kind of data, defined as a set of fields, provides enough flexibility to capture most types of compound datasets that occur in practice [30]. Combining multiple modality sensors such as gyroscopes, magnetometers and accelerometers with other environmental sensors is a good example of such data structure type.

III. SIMILARITY MEASURES AND FEATURE EXTRACTION FOR TIME SERIES DATA
Large time series data requires adequate preprocessing to gain an appropriate approximation of the underlying data representation. The aim of feature extraction is to generate a higher-level abstraction which represents the data while preserving the shape characteristics of the original data during dimensionality reduction. There are several dimensionality reduction techniques specifically designed for time series which exploit the frequential content of the signal and its usual sparseness in the frequency space [97]. In general terms, choosing the distance measure is important and assists in dealing with outliers, amplitude differences and time axis distortion. Furthermore, choosing important features in the data requires sufficient communication of knowledge from domain experts. Thus, the quality of mining approaches is significantly affected by the choice of similarity measures and feature extraction techniques to obtain relevant knowledge from the data. Similarity measures and feature extraction techniques used in the surveyed visual analytics papers are summarized in Table 1.

A. RAW DATA SIMILARITY
Most mining approaches often utilize the concept of similarity between a pair of time series. While dealing with time series data, efficiency and effectiveness are the main targets of representation methods and similarity measures [98]. Tornai *et al.* [99] argue that the distance between two sequences as a measurement plays an important role in the quality of clustering and classification algorithms. The accuracy of such algorithms can be significantly impacted by the choice of similarity measures. Yahyaoui and Al-Mutairi [14] and Wang *et al.* [98] present a comprehensive review for time series measures, classifying them into four major categories: lock-step measures (e.g. Euclidean distance and Manhattan distance), elastic measures (e.g. longest common subsequence [LCSS] and dynamic time warping [DTW]), pattern-based measures (e.g. spatial assembling distance [SpADe]) and threshold-based measures (e.g. threshold query based similarity search [TQuEST]). Pattern-based

measures and threshold-based measures are out of this work's scope as they are not used in the surveyed visual analytics papers.

Euclidean distance (ED) is a commonly used metric for time series. It is defined between two-time series X and Y having length L; therefore, the Euclidean distance, between each pair of corresponding points X and Y, is the square root of the sum of the squared differences [100]. Thus, the two time series that are being compared must have the same length, and the computational cost is linear in terms of temporal sequence length [101]. Along the horizontal axis, the distance between two-time series is calculated by matching the corresponding points [102]. The Euclidean distance metric is very sensitive to distortion and noise [13], and it is not able to handle one of the elements being compressed or stretched [83]; therefore, this approach is not reliable, especially when computing similarity between time series with different time durations [103].

Dynamic Time Warping (DTW) is another distance measure that is proposed to overcome some Euclidean distance limitations such as non-linear distortions. In DTW, the two-time series do not have to be the same length, and the idea is to align (warp) the series before computing the distance [13]. However, two temporal points with completely different local structures might be mistakenly matched by DTW. This issue can be addressed by improving the alignment algorithm, e.g. shape dynamic time warping. It considers point-wise local structural information [104].

Due to its quadratic time complexity, DTW does not scale very well when dealing with large datasets. In spite of this, it is widely used in different applications, such as in bioinformatics, finance and medicine [105]. DTW has several local constraints, namely boundary, monotonicity and continuity constraints [103]. Moreover, some common misunderstandings about DTW are that it is too slow to be useful and the warping window size does not matter much; Wang *et al.* [98] and Mueen and Keogh [106] have attempted to correct these notions. Kotas *et al.* [107] have reformulated the matrix of the alignment costs, which led to a major increase in the noise reduction capability. Other surveys review distance measures such as Euclidean Distance (ED) [108], Dynamic Time Warping (DTW) [109], [110], and distance based on Longest Common Subsequence (LCSS) [98], [111].

Correlation is a mathematical operation which is widely used to describe how two or more variables fluctuate together. Different types of correlation can be found by considering the level of measurement for every variable. Distance correlation can be used as a distance measure between two variables that are not necessarily of equal dimension. In time series data, it is used to detect a known waveform in random noise. Unlike DTW and LCS, correlation also offers a linear complexity frequency space implementation in signal processing [83], [112].

Cross-correlation is the correlation between two signals which shape a new signal, and its peaks can indicate the similarity between the original signals; it is used as a distance

metric [12]. However, cross-correlation can be carried out more efficiently in frequency domain [112]. Autocorrelation occurs when the signal is correlated with itself, which is useful for finding repeating patterns [83]. Walker *et al.* [83] demonstrate that cross-correlation is a slow operation in time series space, but it corresponds to point-wise multiplication in frequency space. It is also considered as the best distance measure to detect a known waveform in random noise. When processing the signal, the correlation has a linear complexity frequency space implementation which cannot be achieved by DTW.

From a data mining perspective, Aghabozorgi and Shirkhorshidi [12] state that Euclidean Distance and DTW are the most popular distance measures in time series data; however, Euclidian Distance is the most widely used distance measure in the surveyed visual analytics papers e.g. [34], [43], [44], [50], [52], [53], [56], [57], [59], [60], [66], [67], [69], [70], [75], [81], [85], [88]–[91], [93], [95], [96], [113]–[115] as it is the most straightforward distance measure compared to others. DTW has only been used in [48], [53], [56], [79] to calculate the similarity of time series data, and papers [34], [35], [61], [72], [83], [85], [86] use correlation and cross-correlation in their works.

B. FEATURE EXTRACTION

Feature extraction is a form of dimension reduction which helps to lower the computational cost of dealing with high-dimensional data and achieve higher accuracy of clustering and classification [116]. Matching features from time series data should be extracted before applying learning algorithms to the vector of extracted features. Several feature-based techniques have been proposed to represent features with low dimensionality for time series data. Wang *et al.* [98] list several methods for reducing time series dimensionality as feature extraction, including Discrete Fourier Transformation (DFT), Discrete Wavelet Transformation (DWT), Discrete Cosine Transformation (DCT), Single Value Decomposition (SVD), Adaptive Piecewise Constant Approximation (APCA), Piecewise Aggregate Approximation (PAA), Chebyshev polynomials (CHEB), and Symbolic Aggregate approXimation (SAX). The types of methods we discuss below are intended to provide examples of popular feature-based techniques, not to define a rigid taxonomy of methods.

Principal Component Analysis (PCA), as an eigenvalue method, is a technique which transforms the original time series data into low-dimensional features. As a feature extraction method, PCA is effectively applied to time series data [117]–[120]. PCA [4] transforms data to a new set of variables whose elements are mutually uncorrelated, thus learning 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 and preserves the most significant. In the surveyed visual analytics papers, [41], [50], [54], [70], [84], [87], [91], [96] use PCA to reduce high-dimensional data and analyze the similarity of the time series data. PCA is a linear dimensionality reduction technique.

Multidimensional Scaling (MDS) is a very popular non-linear dimensionality reduction technique that is useful for effectively representing high-dimensional data in lower dimensional space. This technique has been used in the surveyed papers [36], [48], [54], [56], [57], [63], [78], [81], [84]. MDS is a useful technique which effectively represents high-dimensional data in lower dimensional space; however, it struggles to separate k-Means clusters [84]. Jeong *et al.* [36] use MDS to gain a better understanding of gene interactions and regulatory behaviors. Thus, two different MDS representations are considered with respect to the time series data. One representation shows local differences among genes in the same cluster group while the other shows global differences among all genes in all the clusters. It is also used to reveal the distributions of the time series data, helping to visualize the relations among time series [48].

Transforming time series data into a set of features cannot capture the sequential nature of series. k-gram is an example of a feature-based technique that aims to maintain the order of elements in series using short sequence segments of k consecutive symbols [14]. **k-grams** [121] represent a feature vector of symbolic sequences of k-grams in time series data. Given a set of k-grams, this feature vector can represent the frequency of the k-grams (i.e. how often a k-gram appears in a sequence). It has only been mentioned in [47], [92].

Discrete Fourier Transform (DFT) and **Discrete Wavelet Transform (DWT)** are rarely used in the surveyed visual analytics papers [38], [72], [82]. However, these techniques are used in the data mining field and achieve good results, encouraging visual analytics researchers to adopt these techniques in future research. Discrete Fourier Transform (DFT) is one of the most common transformation methods [1]. It has been used to transform original time series data into low dimensional time-frequency characteristics and index them to obtain an effective similarity search [122]. DFT is used to perform dimensionality reduction and extract features into an index used for similarity searching. This technique has been continually improved and some of its limitations have been overcome [108], [123], [124].

Discrete Wavelet Transform (DWT) has also been used as a technique to transform original time series and obtain low-dimensional features that efficiently represent the original time series data [99], [125]. Chan and Fu [126] use Haar Wavelet Transform for time series indexing, which shows the technique's effectiveness with regards to the decomposition and reconstruction of time series. With a large set of time series data, analysis tasks would face certain challenges in defining matching features; therefore, taking advantage of wavelet decomposition to reduce the dimensionality of data is beneficial [127]. The classification task can be accurately performed utilizing the discrete wavelet transforms technique [128].

Discretization is usually needed when applying feature-extraction techniques in time series data; however, its use can cause information loss [13]. To address this issue, Ye *et al.* [129] introduce time series shapelets which can be directly applied to time series. This technique is based on comparing the subsection of shapes (shapelets) instead of comparing the whole time series sequences to measure the similarity. A binary decision maker decides whether each new sequence belongs to a class or not. The shapelet classifier has some limitations with a multi-class problem, and to overcome this issue, Ye and Keogh [129] use the shapelet classifier as a decision tree. Xing *et al.* [130] have shown that early classification can be efficiently achieved by extracting the local shapelets features.

IV. TIME SERIES ANALYSIS TECHNIQUES

A. CLUSTERING
Clustering is widely used as an unsupervised learning method. The aim of time series clustering is to define a grouped structure of similar objects in unlabeled data based on their similar features. Consequently, data in one cluster is homogeneous, while the data in other clusters are dissimilar. Features do not provide any information about an appropriate group for its objects, they only describe each object in the dataset, assisting clustering algorithms to learn and extract useful information for their structure. Due to the unique structure of time series data (e.g. high dimensionality, noise, and high feature correlation), clustering time series differs from traditional clustering, consequently, several algorithms have been improved to deal with time series.

Most works involving the clustering of time series can be classified into three categories [12]. The first category is whole time series clustering, where a set of individual time series is given, and the aim is to group similar time series into clusters with respect to their similarity. The second is subsequence clustering, which involves dividing the time series data at certain intervals using a sliding window technique to perform the clustering on the extracted subsequences of a time series [131]. The third category is a clustering of time points based on a consolidation of their temporal proximity and the similarity of the corresponding values. Some points might not assign to any clusters and are deemed as noise.

Clustering algorithms embedded in visual analytics systems have received much attention from both the visual analytics and data mining communities for time series data. Unlike the classification task, this task does not require labeled data; therefore, the data is partitioned into groups of similar objects. Most of the existing works that perform time series clustering usually fall in one of the previously mentioned categories. Projection-based methods have received a lot of attention because a scatterplot is intuitive and easy to read. Scatterplots can also provide a unified embedding space for visualizing data and their similarities and show the embedded semantic content [132]. Elzen *et al.* [54] propose a projection-based method to explore and analyze the change of dynamic networks by transforming each time-step network into a high-dimensional vector which is then projected onto a two-dimensional space using dimensionality reduction techniques. Dimensionality reduction is performed for each data window separately, which can then be sequentially visualized, obtaining the similarity across multiple time points evolving over time. Therefore, using the projection-based method can assist with clustering similar time series data so that conventional clustering algorithms can be applied to the projected data [54], [75], [84].

We provide a review of the existing time series clustering methods in the surveyed visual analytics papers, along with the research that has been conducted in the data mining community. These algorithms can be divided into five methods: partitioning methods, hierarchical methods, model-based methods, density-based methods, and grid-based methods. Table 1 summarizes the clustering algorithms used in the surveyed papers. Some papers adapted their clustering algorithms, therefore, an additional section has been introduced in Table 1 to include these clustering algorithms.

1) PARTITIONING METHODS
Partitioning methods are described as a process of partitioning unlabeled data into k groups. The k-Means (KM), k-Medoids (PAM), Fuzzy c-Means (FCM), and Fuzzy c-Medoids are the most popular algorithms for partitioning clustering. Kaufman and Rousseeuw [133] categorize these algorithms into two categories: crisp (hard) clustering methods (including: k-Means and k-Medoids) and fuzzy (soft) clustering methods (including: Fuzzy c-Means and Fuzzy c-Medoids). While in hard clustering methods, each object is assigned to only one cluster, in fuzzy clustering methods, each object is assigned to more than one cluster with a probability. In such methods, the number of clusters must be pre-assigned and most partitioning algorithms cannot tackle the problem of finding the number of clusters [133]. Another issue is that they are not straightforward when dealing with time series of unequal length because of the ambiguity of measuring cluster centers [11].

a: CRISP (HARD) CLUSTERING METHODS
k-Means [134] is a simple and widely used algorithm which divides a set of data into K groups represented by their mean values. After K cluster centers (centroids) are randomly initialized, each example is assigned to the nearest cluster. It iterates until it converges to a locally optimal partition of the data. For each iteration, each example is assigned to the closest cluster center, which will be recalculated based on the mean value of all examples of that particular cluster [135].

k-Means has been used to cluster time series data, achieving efficient clustering results due to its speed, simplicity, ease of implementation, and the possibility to assign the desired amount of clusters [43], [136]. Most of the surveyed papers use commonly applied partitioning methods of clustering, especially the k-Means algorithm [34], [36], [38], [43], [52], [58], [74], [75], [77], [78], [84], [86], [87], [89], [90], [95]. k-Means clustering can be performed on multivariate

time series, where each time point is considered as a vector and the cluster labels are used as symbols to encode the time series [43]. Zhao *et al.* [77], for instance, utilize the k-Means clustering algorithm to cluster visitors based on the time they spend at attractions, thus, it assists to group people in the same cluster if they have similar attraction preferences. k-Means could also be used with visualization techniques, as shown by Wu *et al.* [90], where it is used to determine the most appropriate and reasonable number of clusters for visualization. k-Means has also been adopted in a global radial map to divide all the stations into a number of groups, each having similar change rates [87]. Li *et al.* [86] adopt the k-Means to generate clusters of slopes and map each cluster onto a ring in the global distribution view. In projection-based methods, k-Means is applied to the projected data [75], [84].

k-Medoids or PAM (partition around medoids) [133] is another partitioning algorithm. In this algorithm, a set of k representative samples are initially selected, then each example in the dataset is assigned to the nearest representative sample constructing partitioned clusters. Although this algorithm is like the k-Means algorithm, it is more robust and only differs in its representation. Instead of implying a mean, k-Medoids clusters are represented by the representative data sample in each cluster. This algorithm is often used alongside the DTW distance measure to cluster time series data [137]. Andrienko *et al.* [69] use k-Medoids as a clustering algorithm, which could be better suited than k-Means as it uses medoids instead of means. However, it still has the same issues as the k-Means, where the number of subclusters must be known in advance.

Partitioning has been adopted in some of the surveyed papers such as [41], [53], [54], [56], [59], [69], [70], [72], [81], [92], [96], [114], [115]. Even though fuzzy clustering methods such as Fuzzy c-Means (FCM) or Fuzzy c-Medoids have rarely been used in visual analytics, these methods have promising potential for the future of partitioning methods that can be used in visual analytics systems other than k-Means. Therefore, we briefly highlight these algorithms, along with works that have been accomplished with time series data.

b: FUZZY (SOFT) CLUSTERING METHODS
These algorithms aim to minimize an objective function that usually has numerous undesired local minima [138], allowing fuzzy partitioning instead of hard partitioning. Thus, each sample in the dataset could be assigned to more than one cluster with a membership that measures degrees of association to clusters. Even though fuzzy clustering algorithms are usually more time consuming, they provide more detailed information concerning the data structure [133].

Fuzzy c-Means [139], [140] is the most common fuzzy clustering algorithm and an extended version of k-Means. It provides both effective and significantly meaningful (fuzzy) data partition [141]. This algorithm was later improved by many works [141]–[144]. A dataset is divided into fuzzy groups that differentiate in representatives by minimizing the objective function (within groups) of weighted

coefficients (e.g. distances between objects and cluster center), influencing the fuzziness of membership values.

Fuzzy k-Medoids [145] is another fuzzy partition algorithm which is an extended version of k- Medoids. The candidate medoids are picked (as objective functions located in the cluster centre) from the dataset to minimize all fuzzy dissimilar objects in the cluster.

For time series clustering, unsupervised partitioning has been shown as being efficient in providing good clustering accuracy. Several partitioning clustering approaches (e.g. k-Means [136], [137], [146], [146]–[148], k-Medoids [149], Fuzzy c-Means [142], [150], and Fuzzy c-Medoids [151]) have been used to achieve efficient clustering results for sequences of time series data.

2) HIERARCHICAL METHODS
Hierarchical clustering defines a tree structure for unlabeled data by aggregating data samples into a tree of clusters. It can be used for time series of equal and unequal length [11], [12]. This method does not assume a value of k, unlike k-Means clustering. There are two main kinds of hierarchical clustering methods - agglomerative (bottom-up) and divisive (top-down) [12], [152].

An agglomerative algorithm (bottom-up) considers each object as a cluster, and then progressively integrates clusters. It is the more commonly used algorithm [11], [12] and is involved in many visual analytics works for time series data [85], [92]. The merging process is repeated until eventually, all items are in one cluster or termination conditions are satisfied, such as the number of clusters being sufficient. **The divisive algorithm (top-down)** starts by grouping all objects into one cluster then divides the cluster until each object is in a separate cluster [12], [152]. In their visual analytics system, Bernard *et al.* [91] mentioned two advantages of divisive clustering for time series data. Firstly, the hierarchical structure allows for multiple levels of detail with the same data elements in respective sub-trees. Secondly, the level of detail concept can be achieved with a single calculation. However, both algorithms predominantly suffer from an inability to perform adjustments once a combining or dividing decision has been implemented. Also, they do not have the ability to undo what has been previously done [133], [135], [153], [154].

The basic hierarchical clustering algorithm starts with assigning each vector to its own cluster. Then, it computes the distances between all clusters and saves these distances into a distance matrix. Next, it finds, through the distance matrix, the two closest clusters or objects which will produce a cluster. It updates the distance matrix and returns to the previous step until only one cluster remains [153]. Hierarchical algorithms usually use a similarity or distance matrix to merge or split one cluster, and this can be visualized as a dendrogram [135]. Lin *et al.* [155] present Symbolic Aggregate Approximation (SAX) representation and use hierarchical clustering to evaluate their work. Hierarchical clustering methods can also be divided based on the

way that the similarity measure is calculated; examples include single-link clustering, average-link clustering, and complete-link clustering [135]. CURE [156], BIRCH [157], and Chameleon [158] are some examples for improving the performance of hierarchical clustering algorithms. Hierarchical methods can produce multi-nested partitions that let different users select diverse partitions based on the similarity level that is required. However, it suffers from computational complexity in time and space, and using it to cluster many objects incurs a massive I/O cost.

For visual analytics, hierarchical clustering is often used for classifying time series into separate groups, based on similarities in time series levels [37]. It supports an interactive exploration on multiple levels of detail [52]. Line plots, heatmaps, and dendrograms are the most widely adopted visualization techniques with hierarchical clustering e.g. [34], [37], [39], [46], [50], [52], [60], [85], [89], [91], [92].

The hierarchical method is applied to determine the order of time series data before visualizing and launching interactive exploration [39], [50]. Wijk and Van Selow [159] conducted one of the first pioneer work in visual analytics systems. They use a bottom-up hierarchical clustering approach to identify common and uncommon subsequences that occur in large time series. Then, users can easily interact with the visualization which allows them to select days, find similarities, etc. Battke *et al.* [34] overcame the issue of hierarchical clustering speed for large time series datasets by implementing the rapid neighbor-joining algorithm [160], and then attaching the produced trees to heat-map plots, allowing interactive specialized data exploration.

The hierarchical method creates aggregations which have been visualized as dendrograms, providing multiple levels of detail and an initial overview of similar groups. Visual analytics enhances interactivity, enabling users to change the level of detail by dragging the aggregation level slider [91] or by applying multiple-height branch-cuts to manually select clusters [37].

3) MODEL BASED METHODS

A self-organizing map (SOM), a model-based method developed by Kohonen [161], is a specific type of neural network (NN) that is used for model-based clustering. As an unsupervised learning method, self-organizing neural networks rely on neurons which are coordinated in a low-dimensional (often two-dimensional) structure. Those neurons are iteratively trained by the self-organizing procedure. SOM is one of the most common neural network models and is often used for data analysis. It is also described by Kohonen as an analysis and visualization tool for high-dimensional data [162]. However, SOM can also be used for other applications, such as clustering, sampling, dimensionality reduction, vector quantization, and data mining [163], [164]. The most important feature of SOM is produced in the output layer by the neighborhood relationship [165].

Various extensions have been developed to enhance the SOM's scope and performance, such as adaptive subspace SOM (ASSOM) [166], [167], the parameterized SOM (PSOM) [168], visualization induced SOM (ViSOM) [169], [170], and the Self-Organizing Mixture Network (SOMN) [171]. The SOM uses a collection of neurons usually arranged in a 2-D hexagonal or rectangular grid to shape a discrete topological mapping of input space. At the beginning of the training process, weights are initialized by assigning small random numbers. In this algorithm, each training iteration has three stages. First, an input is presented every time, and then the best matching cell, or winning neuron, is selected. After that, the weight of the winner and its neighbors are updated. The process is repeated until the map converges and the weights have stabilized. In the feature space, the neighboring locations are always represented in the neighboring neurons in the network because they are updated at every step. During the mapping, the topology of the data is maintained as it was in the input space [11], [172], [173].

The self-organizing map (SOM) has been used to analyze temporal data, and is utilized for pattern discovery in temporal data with visual analytics e.g. [34], [44], [45], [70], [79], [91], [114]. Recurrent SOM [174] and Recursive SOM [175] have enhanced SOM for mapping time series data [172]. Fu *et al.* [176] use self-organizing maps to gather similar temporal patterns into clusters. A continuous sliding window is used to segment data sequences from numerical time series before applying the SOM algorithm. SOM also is used in [173] to cluster time series features. In many clustering works, SOM is chosen due to its advantages with regards to certain properties such as parameters selection, data analysis, and better visualization. However, one of its main disadvantages is that it does not work perfectly with time series of unequal length, as it is difficult to define the dimension of weight vectors [11].

Due to SOM being a robust algorithm, Schreck *et al.* [45] use it to render trajectory prototypes and represent data samples on the SOM grid using trajectory bundle visualization. Thus, the trajectory bundles can be visualized at the location of their underlying prototype pattern on the SOM grid. It also organizes the space of movement patterns by arranging prototype trajectories on the SOM grid; this means that neighboring patterns can be compared to each other, and the different patterns smoothly transit over the map. Bernard *et al.* [91] also use the SOM method as a projection technique to make a similarity-preserving color legend for human poses. The grid of the color legend is the result of a SOM that is trained using all feature vectors in the manner of a vector quantization scheme. Thus, the grid structure helps to arrange the most prominent human poses. Moreover, the SOM algorithm can support visualization by representing data on the SOM grid or using the grid of color as a result of the SOM model. The algorithm has also been used in [44] to visually analyze sets of trajectory data which are trained in unsupervised mode. Start and end points of trajectories are indicated over the SOM grid by different colors. The goal of their visualization

is to produce maps of user-preferred trajectory clustering. The surveyed papers have shown that link-nodes and glyphs are the most widely adopted visualization techniques with model-based clustering, e.g. [44], [45], [91].

4) GRID-BASED METHODS
One type of clustering method is the grid-based cluster [35]. This method identifies a set of cells in a grid structure, providing grouped structures in unlabeled data. It is described as a process of quantizing the space into a set of cells made-up a grid. These cells are then used to perform clustering. The fast processing time distinguishes this approach from others. Instead of depending on the number of data objects, they depend on the number of cells in each grid [177]. The two grid-based approaches in [177], [178] are typical examples of efficient clustering algorithms, particularly for very large datasets.

In EpiViz [35], a visual analytics tool for epigenetic features, the grid algorithm is implemented to find similar genes based on the values of their measurements and splits the scatter plot into 5 * 5 cells. Based on their measurements, one cluster of genes per measurement is created for each cell. The scatter plot shows a cluster of genes with their sizes proportional to the number of genes. Thus, it can be said that the grid algorithm, as a machine learning algorithm, assists and interacts with the scatter plot as a visualization technique which provides a classic visual analytics system. Therefore, the EpiViz paper could provide an idealistic model with advantageous features resulting from integrating both machine learning algorithms and visualization techniques to obtain a very effective visual analytics system.

5) DENSITY-BASED METHODS
In density-based clustering, the cluster continues to expand if the density of a set of points with its neighbors is closely packed together, and that cluster is separated by subspaces where the objects have low density. This kind of algorithm is more complex than other clustering algorithms such as partitioning clustering [12]. As it is based on data density, density-based clustering can distinguish noise data and does not require a prior number of clustering, which can be more helpful for non-linear clustering. Landesberger *et al.* [78] highlight some of the advantages of using a density-based clustering technique in their visual analytics methodology for time series data. They state that a density-based clustering is a fast algorithm which does not require pre-setting the number of clusters, is able to detect arbitrary shaped clusters as well as outliers, and uses easily comprehensible parameters such as spatial closeness. DBSCAN [179], OPTICS [180] and LOF [181] are some of the common algorithms that work with the density-based concept.

Aghabozorgi *et al.* [12] state that density-based clustering has not been used broadly for time series data in the data mining community as it has some complexity. However, we found that many of our surveyed visual analytics papers have adopted density-based methods [34], [42], [48], [57], [64], [66], [67], [69], [71], [73], [74], [77], [78], [89]. Looking at combinations of visualization with clustering algorithms, the surveyed papers indicate that the trend is dominated by trajectory data that often adopts density-based techniques for clustering compared to other clustering algorithms. For visualization techniques, maps and the space-time cube are used by Andrienko *et al.* [64], [66], [67], [69], [73], providing mining and visualization techniques which can be applied to trajectory data.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [179] is one of the most highly cited density-based methods. It depends on a density-based concept of clusters which is designed to detect clusters and noise in a set of data. For each point of a cluster, the eps-neighborhood must have a minimum number (minPts) of points. Therefore, the two parameters, eps and minPts, must be known for each cluster or, at the very least, for one point from the particular cluster. In every cluster resides two points, the core and border points, which are on the cluster's border. DBSCAN has good efficiency on large datasets and aims to discover clusters of arbitrary shapes. For example, Chae *et al.* [74] and Zhao *et al.* [77], in both visual analytics systems, use DBSCAN to group visitors into corresponding clusters. Zhao *et al.* [77] utilize the longest common subsequence (LCS) to measure the similarity of two visitors' sequences before applying DBSCAN.

However, DBSCAN cannot transact with clusters of various densities, which is one of the main problems for this algorithm. In contrast, OPTICS (Ordering Points To Identify the Clustering Structure) [180] can deal with the issue of an unknown number of clusters with different densities [182]. Local Outlier Factor (LOF) [181] also shares certain notions with DBSCAN and OPTICS with regards to local density estimation, and depends on distances in its local neighborhood. Most clustering algorithms are developed to find and optimize clustering, and they usually ignore noise when the clustering result is produced, but the LOF tries to assign for each object a degree of being an outlier.

B. CLASSIFICATION
Classification is described as mapping data into predefined classes. The classification task is referred to as a supervised learning method because the classifier is constructed using training data, and classes are known in advance. In this task, the algorithm is trained on dataset examples, and tries to assign each set of data into its appropriate class; in other words, assigning time series patterns to a specific category [13].

In classification, the aim is often to learn what the unique features that distinguish classes from each other are. Thus, when an unlabeled dataset is entered into the system, the classification task can automatically determine to which class each series belongs [183]. The k-nearest neighbors, decision tree, support vector machines and neural network are the most widely used algorithms for the time series classification task. Even though these algorithms have received much attention in

the data mining and machine learning communities, embedding their use in visual analytics systems for time series data is still a relatively young and emerging field. The next sections intend to provide a review of the few existing time series classification in the surveyed visual analytics papers along with the works that have been adopted in the data mining community.

1) k-NEAREST NEIGHBORS (k-NN)

The k-nearest neighbors algorithm performs a straightforward function on data. There is no learning process; in order to produce an output for a new test input x, the k-nearest neighbors to the new sample at test time can be found in the training data, which then returns the major class label, producing an output at test stage [4]. Despite there being numerous classification algorithms that have been used to classify time series, evidence shows that the simple nearest neighbor classification is extremely difficult to beat [184], [185]. Xi *et al.* [186] and Rakthanmanonet *et al.* [187] have shown that the simple combination of one-nearest-neighbor with dynamic time warping (DTW) distance produces good results, but it suffers from computational complexity with the DTW algorithm [186].

2) DECISION TREE (DT) AND RANDOM FORESTS

The decision tree (DT) is one of the most popular classifiers. It is generated by algorithms that identify various ways of dividing a dataset into branches [188]. The tree has three kinds of nodes. In the root node, the outgoing branches can be divided into one or more branches. In the internal node, one incoming branch can be divided into two or more outgoing branches. In the end node, leaf nodes represent classes and branches represent decisions. Starting at the root, the classifier makes decisions to reach the class label [189]. The decision tree can also be utilized under uncertainty as a sample for sequential decision problems. It assists with describing the decisions that will be made, the cases that might happen, and the results that are related to each of the events and decisions.

C4.5 [190], ID3 [191], Classification And Regression Tree (CART) [192], [193], and CHi-squared Automatic Interaction Detector (CHAID) [194] are examples of decision tree algorithms. The complexity of a tree impacts its performance and accuracy. The criteria and pruning method that are used can control this complexity, and certain metrics can be used to measure it. These metrics include the depth of the tree, the overall number of nodes, the number of used attributes and the overall number of leaves. The rule induction always links to the decision tree induction, and every path from its root to its end can be converted to a set of rules [195]. The decision tree's performance is better when it deals with discrete features.

Random Forests are an ensemble of bagged decision tree learners with randomized feature selection. Breiman [196] defines it as a collection of randomized decision trees, thus, it takes the decision tree concept a step further by producing many decision trees. In a random forest, each decision tree is learned from a random subset of features and a random subset of training examples [197], [198]. It can be used for classification as well as regression.

For random forests, the training algorithm applies general techniques of bootstrap aggregating or bagging. In bagging, it trains an often large number of classifiers on random subsets of the training set, classifying by using the majority vote of all classifiers. In boosting, it operates as per bagging, but introduces weights for each classifier based on performance over the training set.

Decision trees (DT), have been adopted in visual analytics with high levels of accuracy. Xie *et al.* [40] use a decision tree in the VAET system which highlights interesting events in e-transaction data. The system uses a probabilistic decision tree learner to estimate the salience of each transaction in a large time series. Then, the saliency values are visualized in a time-of-saliency map. This visualization allows analysts to explore, select and conduct a detailed examination of interesting transactions, displaying them in a new visual metaphor called KnotLines.

3) SUPPORT VECTOR MACHINES (SVM)

Support Vector Machines (SVM) is an effective classification method. It is widely used and has shown substantial achievement in solving sequential time series classification tasks [199]–[203].

The SVM discriminates between positive and negative examples, and through the use of said examples, it learns to classify and produce positive and negative classes [4]. For linear cases, SVM [204]–[206] aims to find a class identity by mapping series into a high-dimensional feature space. Once the similarities between series have been measured, SVM separates two classes and enforces a larger margin hyperplane, which is the gap between classes. Thus, SVM acts as a large margin classifier for accurate classification and efficient generalization.

For non-linear cases, SVM often uses kernel functions, which represent a non-linear decision boundary that separates the positive and negative samples. The kernel function is appropriate with high-dimensional feature spaces and has been applied to measure the similarities between two given time series [201]. Many kernel-based methods corresponding to different measures of similarities and which efficiently overcome time series classification problems have been proposed [201], [207]–[209]. Multiple kernel learning is an optimization problem [201] whose solution has been proposed by [210]. They present an efficient algorithm that solves the multiple kernel learning problem and works with many samples or multiple kernels which need to be combined.

Support Vector Machine (SVM), as a time series classification model, has been integrated with visual analytics systems [82], [94], [113]. This procedure allows scientists and domain experts in such fields (e.g., biology) with a little background in machine learning to build classification models with high

levels of accuracy [82]. Lu *et al.* [113] supported the creation of the SVM model along with two other different types of models, Linear Regression and Multilayer Perceptron, combining feature selection and model cross-validation through numerous interactive visualizations, which help analysts in their building of such a model. Kim *et al.* [94] developed a visual analytics tool that incorporates machine learning algorithms (supported vector machine) to predict coded undesired behaviors.

4) NEURAL NETWORKS (NN)

Neural networks are learning algorithms that mainly rely on statistics. This kind of algorithm learns from data using its own learned features [4]. Neural network algorithms have been efficiently used to solve several tasks. The task of classification, especially time series classification, has received particular attention with regards to using different kinds of neural networks, such as multi-layer perceptron (MLP) [211], convolutional neural networks (CNN) [31], [212], [213], and recurrent neural networks (RNN) [214].

Multi-layer perceptrons (MLPs) represent a type of neural networks that have been used as classifier. Its architecture comprises fully connected layers, and each layer contains neurons with weighted interconnections between them [211] called parameters. Neurons act as switching units associated with weights that are interconnected among them. The aim of this model is to ideally approximate a function (e.g. classifier function) by mapping the input values into a category (a class) learning the parameters (weights) [4]. For time series classification, class labels should be given so that a learning function maps the series into an appropriate class. Thus, the weights are learned by finding the best relationship between time series and their appropriate classes [211]. From the visual analytics perspective, multi-layer perceptrons (MLPs) have been used by Lu *et al.* [113] in their visual analytics system. They use backpropagation and allow users to select which algorithm to use, set the number of folds for the stability test, train models to predict and compare between available models.

Convolutional neural networks (CNNs) are a recently introduced kind of neural networks that have been developed for processing data that has grid-structured topology, such as time series (1-D grid) data and image data (2-D grid of pixels). CNN architecture comprises convolutional layers for spatially related feature extraction and fully connected layers used for classification. Convolutional layers are utilized, as feature extractors, to learn features through mapping the raw data into a feature space, and the trainable fully connected layers perform classification based on the learned features from the convolutional part. The convolutional part generally consists of multiple layers; each layer has three stages: the convolution stage (filter), the detector stage (activation) and the pooling stage [4]. The input and output of each stage are called feature maps [31]. In the training stage, the forward and backward propagation algorithms are used to train the CNN and estimate parameters. A gradient-based optimization method is utilized to minimize the loss function and update each parameter [213].

Unlike 2-D grid (e.g. image data) input, convolutional neural networks for time series uses a 1-D grid, so instead of holding raw 2-D pixel values, the input of time series classification is multiple 1-D subsequences. In this case, multivariate time series [31] are separated into univariate ones so that feature learning can be performed for each univariate series. At the end of feature learning, trainable fully connected layers are adopted to perform classification.

The univariate time series are considered as input that is fed into the convolutional layers, learning features through convolution, activation and pooling layers. The 1-D convolutional layer extracts features by applying dot products between transformed waves and a 1-D learnable kernel (filter) [215], computing the output of neurons that are connected to local temporal regions in the input. This stage is followed by the activation layer, which is used to perform non-linearity within the networks, allowing learning of more complex models [216]. In the pooling layer, a down-sampling operation is performed to reduce the resolution of input time series [31], which in turn reduces complexity and generalizes features in the spatial domain. After extracting feature maps from multiple channels, they are fed into other convolutional layers and then pass them as inputs of the fully connected layer. In the fully connected layer, the class score will be computed, where each of the result numbers corresponds to a specific class.

Time series classification faces some obstacles and difficulties, such as feature representations at different time scales, and can be distorted by high-frequency perturbations and random noise in time series data [215]. Several multi-channel CNN architectures have been used for the task of time series classification [31], [212], [213], [215], [216]. The results of all adapted CNN classifiers are competitive for both classification accuracy and performance with regards to overcoming the challenges.

The classification algorithms applied in our surveyed papers are usually embedded in visual analytics systems [40], [82], [83], [94], [113]. The k-nearest neighbors, decision tree, support vector machines, and neural network are used in some recent works, but are not as common as clustering techniques.

V. VISUAL ANALYSIS
A. VISUALIZATION TECHNIQUES
Visualization transforms symbolic data into geometric data [217]. The result of this process can help people to understand the data by presenting it in a graphical format, helping users or analysts to observe, analyze, make decisions, and identify patterns and correlations based on visualization. The visualization can also help to detect and see information and relations between data which might not be recognized when looking at numerical data [218]. In this way, it can aid scientific discovery and enhance the likelihood of gaining deep and unexpected insights, which sometimes leads to new hypotheses.

At a basic level, time series data (e.g., from sensors) is presented in 1-D charts, with multiple sensors displayed on the same chart or linked charts. Different visualization techniques (ripple, stacked, river, stream) and interaction techniques (zoom, pan, select) allow the user to select the time duration and obtain visual feedback. Interaction with the linked view will highlight regions in the time series and any pattern recognition techniques will highlight data in the time series, helping to understand and analyze data over time [38], [48], [59], [86], [90], [93]. With stacked, river, and stream graphs, each item is displayed as a colored current whose height changes continuously as it flows through time. The overall shape comprises all the items considered, and it can provide an overview of the topics that are important at points in time. Various possibilities for interaction are used, which allow users to browse and zoom into details of the time duration, as well as to select from the shape.

For time series data, achieving a good visualization helps users not only to create interesting images or diagrams, but also to amplify cognitive performance. Thus, visualization should communicate with the mind to simplify the data complexity. Aigner *et al.* [15] present three main criteria, these being expression, relevance, and effectiveness, that need to be satisfied in order to achieve a good visualization, exploiting both human visual perception and huge computer processing.

In this survey, visualization techniques are divided into nine categories. These classifications draw from the comprehensive vocabulary of the visualization taxonomy presented by Borkin *et al.* [219]. This taxonomy is used and modified to include all visualization techniques that are used in our surveyed papers, which are summarized in Table 1. They include the following: **Bar** (Bar Graph, Ripple Graph, Histogram), **Area** (Stacked Graph, River Graph, Stream Graph), **Circle** (Pie Chart, Radial Chart), **Line** (Line Plot, Parallel-Coordinate, Time-of-Saliency, Knot-Lines), **Geographic Maps**, **Grid and Matrix** (Heat Map, Ranking View, Calendar Map, Space-Time Cube, Tessellation), **Point** (Pixel, Bubble Chart, Scatterplots), **Trees and Networks** (Dendrogram, Node-Link), and **Glyphs**.

From the surveyed papers, it can be noticed that while some techniques dominate others, they share the same goal, which is to present as much information as possible in the display to the user. Thus, there is a wide pool from which to select visualization techniques that can smoothly deal with big data in order to reduce data size and produce a visualization structure which allows the user to explore, analyse, and understand the data.

In the same context, Table 1, shows an increasing trend of using a variety of visualization techniques with time series data. Also, line plots, geographic maps, heat maps, histograms, and bar graphs are the most commonly used techniques in the surveyed papers. Most of them are used to give an overview of the dataset by displaying the time-dependent relations of actions. In contrast, some visualization techniques are rarely used, such as tessellation and streamgraphs,

while some are presented as new visualization techniques such as time-of-saliency and knotlines.

B. INTERACTION TASKS

Visual analytics merges machine and human capabilities to facilitate exploration, analysis, understanding and provide insights of exploratory analysis for data and methods. Visual analytics present the chance for analysts, through interaction tasks, to analyze, explore, reason, discover, and understand important structures in complex data and architecture of methods [20]. Thus, users can be involved in the process through interaction tasks providing directed feedback to the system.

Early steps in visual analytics were investigated by Tukey [220] on exploratory data analysis, encouraging to support direct interaction with data. Following this work, numerous interaction methods have been developed to support various types of analysis data and methods, assisting users and analysts to better understand, explore, analyze, and gain insights. Researchers in the field of visualization have made efforts to benefit from user interactions in order to achieve analytical reasoning and integrate users into a comprehensive visual analytics system [7]. Several works for different visualization tasks and interaction methods have been presented. Those existing works can be classified into three categories, namely low-level tasks, or interactions (e.g. [218], [221], [222]), high-level tasks (e.g. [223]–[225]), and multi-level tasks (e.g. [226], [227]).

In this work, we utilize a typology of abstract visualization tasks by Brehmer and Munzner [226]. Their typology provides potential for rigorous analysis as it does not only focus on low-level tasks and high-level tasks, but also addresses the gap between them; these tasks are termed as multi-level tasks. This typology allows us to better interpret our survey from an interactive visual analytics perspective, given that it provides multi-level visualization tasks and a straightforward way of describing complex tasks as linked sequences of simpler tasks.

They identified six main multi-level tasks which are related to visualization tasks in the surveyed papers. We briefly summarize each task with all its subtasks and comment on how they are used in the surveyed papers. In the high-level task (**analysis**), users or analysts can analyze data using visualization tools so that they can **consume** information in many domain contexts or **produce** new information using available resources such as existing data elements. In the mid-level task (**search**), users or analysts can search elements of interest using visualization tools. The search task is classified into four types: **lookup**, **browse**, **locate**, and **explore**. In the low-level task (**query**), the users or analysts already found targets, thus, they can **identify**, **compare**, or **summarize** the pre-found targets. The visualization tasks in our surveyed papers are summarized in Table 1 under the headings: **Analysis**, **Search**, **Query**, **Encode**, **Manipulate**, and **Introduce**.

From the surveyed papers, it can be noticed that low-level tasks are more commonly used than high-level abstract tasks.

As shown in Table 1, using visualization tools as high-level tasks to analyze data is rarely done in the surveyed papers. In contrast, low-level tasks are often used; for example, query tasks are often used to find targets. Selection and navigation interaction methods are also widely used to provide a range of different options which can be applied to any element in visualization systems. Moreover, the filtering method is frequently used when individual view of sequence data needs to be filtered.

C. VISUALIZATION AND ANALYSIS TECHNIQUES

We have conducted the review from the perspective of the data mining and visualisation communities and how the two integrate to produce visual analytics systems. The data mining community utilises visualisation to a lesser extent and with the specific goal of demonstrating the efficacy of methods under research. Images are intended to be static figures, there are many examples of using t-SNE (clustering) overlaid with colour to represent classification to convey how well a new technique performs or how well a data set can be processed. Another example is that of utilising heat maps to indicate which features from training sets contribute to the model classifier.

Visual analytics provides different perspectives and goals to satisfy the user demands. Interaction becomes a key goal where the system should impart more knowledge through the capability to interact with data or model parameters. This can lead to a different emphasis on the methods chosen to process the data. An effective clustering algorithm such as t-SNE led to ineffective user interaction because of spatial inconsistency after reduction to 2D used in the creation of the interactive user interface [131]. Alternative clustering techniques PCA and UMAP projected similar data to similar spatial locations in 2D (Fig. 3). Feedback in user studies and from domain experts indicated that the latter dimension reduction approaches are more suited for deriving user interfaces [131].

Parallel coordinates is a familiar interaction tool in the visualisation community to enable the exploration of high dimensional parameter spaces, but we saw no use of parallel coordinate visualisations as static images in the data mining literature. Primarily this is due to it being a useful tool to interrogate data when interaction is employed. Each axis can represent a parameter in the model or clustering approach, etc. allowing the user to experiment a gain feedback through alternate views [228](Fig. 4). Indeed, the utility of these approaches is through multiply coordinated views where direct interaction in any of the views highlights the same selection in each view space (Fig. 3).

For temporal data with a spatial component, a common processing approach is for locations to be quantized, and paths through the quantized locations creating a motif which can be matched using similarity measures. 1-D curve similarity measures are employed directly on the data. Multiple sensors, weighted similarity, or higher-dimensional data is reduced in dimensionality (PCA, MDS, *non-linear DR*) before clustering. Similarity measures include Euclidean
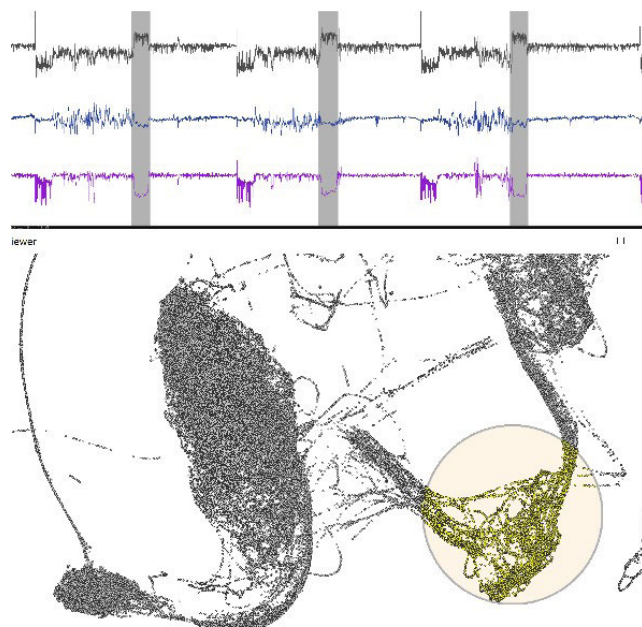

FIGURE 3. UMAP clustering of time-series animal behavioural data leads to consistent neighbourhoods in the 2D interface (compared to t-SNE which does not). Also shown, a k-nn cluster and how pattern matching in multivariate data is achieved through the interface [131].
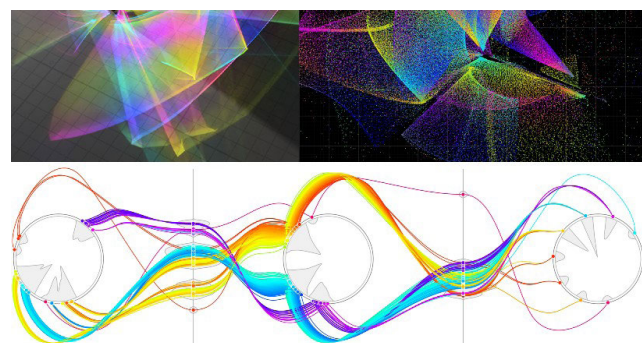

FIGURE 4. Parallel coordinates plot with annular and linear axes, colour coded splines representing the data, and density plots on the annular axes. The view is coordinated with the (PCA dimension reduced) point data (top right). A density rendering based on the data is given (top left). User interaction is principally through the parallel coordinate plot to isolate the overlapping manifolds in the data [228].

[34], [43], [44], [50], [52], [53], [57], [59], [60], [66], [67], [69], [75], [85], [88]–[91], [93], [95], [113], DTW [48], [53], MDS [36], [48], [54], Pearson, Fisher Discriminant [88], cross-correlation [83], etc. Through the SOM algorithm, as model-based clustering, the SOM grid provides trajectory bundle visualization of locations underlying prototype patterns, allowing experts to visually analyze sets of trajectory data [45] or a search interface, meaning that analysts can control clustering [91].

If each curve/path has similarity computed against all other paths, the result is a symmetric square matrix where each entry represents the degree of similarity/dissimilarity. We can employ clustering techniques such as agglomerative clustering and DBSCAN to create a hierarchy which can

be displayed as a tree structure (dendrogram [91]). Cuts can be taken through the tree to simplify the data. The tree provides a useful interaction interface to update and query results in the other linked windows. Dendrograms as a static image infrequently appear in machine learning literature, but again they create a useful interactive tool since a cut through the dendrogram can produce a specific instance of a visualisation representing different levels of clustering (or data aggregation).

Time-based (e.g., one hour, one day), or pattern-based (e.g., recognising a pattern using a variety of similarity measures or change detection) can result in data segmentation. The segmentation results are visualised or used as input for further processing steps [53]. Users can influence segmentations indirectly through choices concerning the segmentation algorithm, changes in its parameters, or by direct selection and labelling of the data.

Visualising segmented data offers significant visual cues for determining outliers or clusters of data. For 1-D data, multiple segmented data can be plotted on charts (and multi-dimensional on linked charts). Trends, clusters and outliers can be detected visually [59]. Interaction can allow brushing in the chart to remove, select, label or highlight groups of associated data. Queries can be generated using slope tools or ranges, and curves will either match or not match such queries. These queries can be stored for future use to act as triggers or stored procedures on the data.

Apart from 1-D charts, another main approach is to use radial depiction of data. The data can be visualised as line or bar charts in a circle with the x-axis around the circle, and y-axis away from the circle. Typically, the x-axis represents time, with multiple axes radiating from the centre indicating durations (e.g., hours in a day, days in a month, or months in a year). Transformed data may place spatial coordinates on the x-axis (with a map central to the visualisation), and the y-dimension could be time, with distance from the y-dimension then indicating further attributes such as intensity (of the sensor – e.g., pollution levels [86], [89], and shells of data appear around the circle (stacked/river charts). Multiple small versions would create glyphs, or a single view linked to other views, which offers more detail.

Calendar views [53], [78] also offer successful interaction, allowing visualizations to aggregate according to the days selected. Selections can involve months, a certain day of the week and workdays versus weekends. A secondary view based on the above chart or network views can offer focus-and-context associating the detailed view within the overall context of the annual view. The calendar view utilizes colored patterns to indicate different clusters; therefore, the selected elements become active and bigger, which cause unselected elements to become smaller. Differing from radial plots, the calendar view allocates the same amount of screen space to individual patterns, giving them equal visual importance [53], allowing to visualize during which time stamps the temporal clusters occur.

The similarity matrix also serves a useful visualization and interaction tool, and is displayed using color mapping e.g., resulting in a heatmap [52], [94]. Rows and columns can be sorted to reveal patterns. Individual selections in the heatmap highlight data pairs in the source data. Larger selections highlight groups of data with the degree of similarity chosen. Selections are linked to other views of the data. Sorting can also be applied to any of the other linked views, e.g., multiple bar charts can be sorted by decreasing similarity from a user-selected pattern [83], [85]. It can also be used for network graph. Different colors and pixels are used to represent the data, emphasizing the relationships between elements. The similarity matrices explain to which degree the clustering would change for the next parameter setting. In this kind of visualization, the user can select a similarity threshold and algorithm which helps to perceive the dynamic network from different perspectives.

With regards to graph/network data, networks are directly visualized as node-link diagrams resulting in clutter [52], [54], [78]. Standard techniques are used to simplify the graphs, such as using edge bundling, weighting edges according to the linkage, or higher order curves to emphasize path connectedness. Node-links can be converted to matrix view with each matrix element storing the edge weight between the two nodes. The matrix can be visualized directly (with edge weight mapped to color). The network view provides an overview of the clustered nodes which have a similar behavior over time and edges reflect connections between these clusters.

Visual analytics systems offer direct views of the data (e.g., visualisation of the raw accelerometry data (Fig. 3 top) or abstract views (Fig. 3 bottom) where data has undergone processing such as dimension reduction to create the interactive interface. Throughout our study the essence of visual analytics is to provide multiply coordinated concrete and abstract views of data. This allows interaction with parameter spaces to enable human cognition to play a vital role in information and knowledge discovery.

Analysts usually change their exploring strategies and switch between analytical techniques and visualizations to collect different findings. However, these analytical techniques (black-box methods) might confuse the end-users or provide results that do not lead to a solution to the problem, and some of them require user action such as k-means requiring the assignment of the number of clusters. To be beneficial in visual analytics, the analysis techniques should be fast enough in terms of response for efficient interaction, parameters of the analytical technique have to be representable and understandable utilizing the visualizations and parameters have to be adjustable by visual controls [20].

There are numerous challenges associated with visual analytics system usability and process understanding. To obtain more confidence, the user should be aware of the source of data and the transformations that have been applied on its way through the processing stages (e.g., preprocessing, analysis tasks and visualization techniques). Rapid feedback

is significant in visual analytics interfaces, and that represents challenges to various of the domains related to visual analytics. Due to the complexity of human interaction, evaluating visual analytics systems is especially complicated, and integrating machine learning algorithms in to these systems adds additional complexities and opens questions such as how the model succeeds or determines what a good solution is, why a model predicts a value, or why a model provides a classification label which are sometimes beyond of the scope of interactive visualization issues. Some works such as [229]–[231] shed light on the black boxes of classification and clustering algorithms and explain the determined decisions which assist to understand these algorithms and enable the comparison of different prediction methods. These questions are very important in order to understand the model outputs and provide appropriate visual representations and interaction techniques.

VI. EVALUATION APPROACHES

A systematic evaluation, controlled by a set of standards, identifies and validates the degree of achievement or value of proposed systems, techniques, methods and algorithms. Since the space of visualization systems design is massive, Munzner [227], [232] subdivided this complex problem into four sequential layers that separately solve various concerns, presenting a nested model for visualization design and validation. At the top level, details of a specific application domain are considered. Next is the design of data and tasks abstraction. The following level concerns the design of visual encoding and interaction, while the last level involves the design of algorithms.

This research utilizes Munzner's work [227], [232], which presents different appropriate evaluation approaches at each design level, including **field / case studies, controlled lab / user studies, usability studies, heuristic,** and **algorithms performance**. These approaches were applied to our surveyed papers (summarized in Table 1). At the top level, field studies or case studies form the most common evaluation approach, where investigators gather qualitative data through semi-structured interviews and observing people's actions in real-world settings. At the abstraction level, studies or case studies are also used as qualitative validation to evaluate a member of the target users by observing and documenting their use of the deployed system. At the visual encoding and interaction idiom level, a controlled lab study or user study is used as an evaluation approach. Through this, quantitative measurements (e.g. time, errors, quality, and preferences) are collected as well as qualitative measurements (e.g. questionnaires and qualitative discussion). Also at this level are usability studies, another qualitative evaluation approach which aims to prove that the deployed system is usable. Heuristic evaluation is another, quantitative and qualitative measurements, validation approach that involves experts in the field to ensure that the visualization design does not violate any guidelines used to justify the usability of a visualization system. At the algorithms design level, the quantitative

evaluation approach is used to validate the performance of algorithms such as their speed and computational complexity.

The evaluation approaches which have been applied to our surveyed papers are classified into five categories adopted from Munzner's work [227], [232]: case studies, usability studies, controlled user studies, algorithm performance and others. Table 1 summarizes each evaluation approach used on a surveyed paper, classifying based on years. It should be noted that the case study approach is most commonly used in the surveyed papers compared to other evaluation approaches. We have also noticed from our survey that other evaluation approaches are also used, such as ground truth [52].

VII. INTEGRATION OF VISUALIZATION AND ANALYSIS TECHNIQUES

A. MODEL BUILDING VISUALIZATION

The above processes, visualizations and interactions can result in a large corpus of labelled data suitable for visual and statistical interrogation. Additionally, labelled data is useful for model building, using data mining approaches as discussed earlier. Such models can be used to aid the user with further segmentation and labelling of the data [40], [57], [83], building predictive models for the future [113], and identifying patterns and behaviour of systems or individuals in the data [94]. By exposing algorithm choice through the interface, along with parameters, the user can play an interactive role in deciding the best approach for their data [44], as effective algorithms for time series analysis always require precise choices of approaches and parameters in order to be able to solve clustering and classification tasks. We notice from our survey that several interaction methods are not specific to data only, interestingly, a variety of interactive tools are combined to support analysts in algorithm selection (e.g., [89], [113]), training (e.g., [44], [113]) and testing (e.g. [113]). Moreover, several systems interactively provide analysts with a variety of controlling options for time series analysis tasks such as control algorithms parameters (e.g., [38], [44], [52], [78], [94]) or control threshold (e.g., [35], [37], [40], [44], [52], [83], [88], [95]). For example, overplotting matched time-series data leads to new interfaces where direct data selection can accept or reject data from the view without the need for further model training. Exposing model parameters to the user allows the understanding of their inter-relationships and how they impact the algorithm (e.g., feature detection) performance (Fig. 5).

Visual feedback of the model using the visualizations and interaction results in the effective capture of domain knowledge, fulfilling the definition of visual analytics and including humans in the loop. The models range from pure clustering, such as clustering patients on medical records, which can lead to predictions about how an individual patient's condition will evolve [46], to utilizing classification techniques such as SVM [82], [94]. Such models can be used to aid the user with further interactive clustering while representing data samples as discussed earlier.
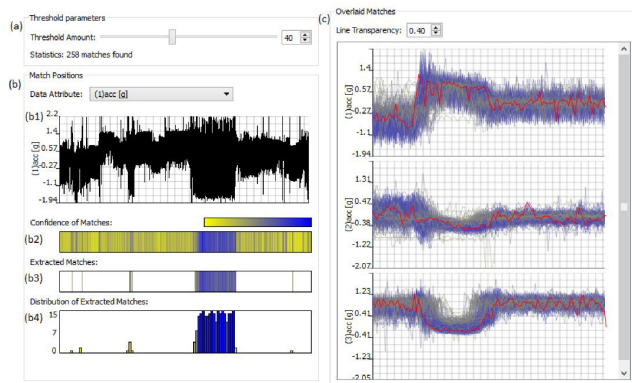
FIGURE 5. Cross-correlation in frequency space is used to find matching time-series patterns with low computational complexity. (a) The user can interact with the cross correlation threshold, (b) and in a linked view see where the matches occur in the overall time-series. (c) Overplotting allows the user to inspect matching patterns. The cluster centre is plotted. The user can interact with the view in (c) to select and "delete" undesirable matches [83].

Choosing the analysis algorithms such as k-means, hierarchical clustering, or the self-organizing map or giving feedback during the analysis process such as k-means which requires the user to specify the number of clusters as input are some examples of the interaction between the end-user and visual analytics system. Therefore, the implication of visual analytics and the goals of the end-users on the choice of analysis algorithms are fundamental and require further investigation in terms of what kind of visual controls are required to manage the algorithm and assess the quality of the proposed solutions side by side with the interactive visual representations.

B. EMERGING TRENDS

The merger of visual analytics and machine learning offers many potential opportunities for time series data analysis. However, a large effort is still needed at the algorithmic and software levels to help embed fast machine learning techniques in visual analytics systems. From a performance perspective, dealing with massive datasets in terms of quantity and speed of data to be visualized and interacted with in real-time is crucial in visual analytics systems. Therefore, response times are very important and such factors can play a major role in an interactive visualization. Thus, developing fast machine learning for interactive visualization is one of the open research topics associated with integrating the two domains.

Also, one of the major technical barriers is that the existent software tools are highly divided between these two domains; for example, visualization tools are often written using programming languages like C++ or using libraries such as d3js (a JavaScript library), which are powerful with regards to maintaining close control over the visualization technique and user interaction. On the other hand, most of the advanced machine learning algorithms are usually written using different libraries in statistical or programming languages such as

Matlab, R, or Python (Machine learning libraries like scikit-learn, TensorFlow, etc.), where they aim to learn complex models from (often large amounts of) data but provide limited interactive information visualization. Therefore, there is an urgency to find a standard software environment which can be used to assist visual analytics developers with integrating machine learning techniques effectively and efficiently in interactive visualization systems [10].

Recent visualization research has seen and increased use of sophisticated algorithms, especially in projection-based methods which have a stochastic nature [54],Ali2019. Thus, the outputs of these algorithms may rely on different settings, e.g. random initialization, which sometimes have major effects on results and evaluations. These algorithms should be measured in terms of their robustness, generalizability, stability analysis, sensitivity analysis, etc. The robustness of an algorithm concerns its ability to handle any kind of input. An algorithm's generalizability sigifies that it can be generalized into a greater dataset (unknown data) than the dataset (small known data) used in the training process. Stability analysis refers to the analysis of errors in numerical computation (if the errors are increased, the algorithm is numerically unstable, and if the errors are abated, the algorithm is stable). The sensitivity analysis of algorithms involves analyzing the alteration of outputs with respect to the inputs. Therefore, visual analytics developers must take into account these factors alongside others which may have major effects on visualization results [7].

Moreover, some machine learning algorithms embedded within visual analytics systems for time series data are still part of a relatively young and emerging field, even though they have received wide attention in the data mining and machine learning community. To mention but a few, Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) have rarely been used as dimensionality reduction techniques by visual analytics researchers, while these techniques achieved good results for time series data in the data mining field. Moreover, some clustering algorithms, such as the fuzzy clustering methods, and classification algorithms are currently under-represented in visual analytics works but are successful in the data mining community, and are therefore something that visual analytics researchers should include in their future works.

There are several challenges which we perceive as interesting research directions for combining machine learning and visualization techniques. Firstly, there is no existing unified or systematic solution to support the user, which explains the scarcity of classification algorithms used in the surveyed papers. Secondly, there is a visualization challenge in terms of clarifying the reasons behind why such algorithms demonstrate impressive classification performance.

One interesting potential research direction of combining the two fields of machine learning and visualization techniques is building user-driven algorithms specifically geared for a visual analytics approach to overcome difficult challenges for time series data. Involving users in the process,

through interactive methods, allows them to provide directed feedback to the system. Formulating a user-centric approach through combining automated analysis methods and interactive visualization is an efficient approach to visual analytics. This puts emphasis on the visualization community to apply visual interfaces to existing algorithms provided by the data mining community.

Deep learning algorithms (e.g. CNN, RNN and LSTM) are often perceived as black-box models due to their ambiguity and unclear working mechanisms [233]. Although these algorithms have been used for time series data by the data mining community, there is little work on CNN, RNN or LSTM with visual analytics. This leads to the other interesting potential research direction of combining the two fields, as there is no clear understanding of why deep classification algorithms achieve highly performant results when solving such a task. Thus, visualization techniques are needed to explore such complex models as well as illustrate and explain their internal operation and work mechanisms. This would allow to gain general insights and obtain an overview of how to control and improve such models. Efforts have been made in the field of computer vision to clarify the learned features of deep learning algorithms on image data. The existing methods of previous works can be categorized into two different groups: code inversion (e.g. [234]–[236]) and activation maximization (e.g. [237]–[239]). In the field of visualization, a set of visualizations have been developed to help machine learning experts clearly understand such deep complex models (e.g. [240], [241]). Liu *et al.* [240] have recently presented an interactive visual analytics approach which allows for the better understanding, diagnosis, and improvement of deep CNNs.

VIII. CONCLUSION

This research is considered a comprehensive survey for time series data, focusing equally on both machine learning and visualization from the visual analytics perspective. Time series data can be obtained from different sources which have been categorized into four types based on the surveyed papers. During research, we focused on two mining tasks; clustering and classification. At the beginning, we review both tasks from the data mining perspective. They achieve great performance and accuracy when dealing with time series data. This success led us to review a promising field where both automated analysis techniques and interactive visualizations can be combined to easily understand, explore and analyze large and complex datasets. We cover over 60 papers in detail, which were selected with the criteria that every paper must involve time series data and visual analytics, using either clustering or classification tasks. It can be noticed from the surveyed papers that many visual analytics works use clustering more than classification. Because of a lack of label data, keeping humans in the analysis loop is paramount in order to help users adjust and explore the influence of different clustering choices during the analysis process. Visualization and interaction techniques are also surveyed in

the reviewed papers and classified based on previous literature. Such classifications have been modified and changed to be compatible with the surveyed papers. The evaluation approaches of every paper were also studied and categorized. As a result, researchers can use this review as a guide for new investigations. In the end, we believe that this paper is a starting point towards clarifying the major concepts that have been presented, and provides a valuable guide to the emerging field of integrating data mining techniques with visual analytics.

REFERENCES

[1] T.-C. Fu, "A review on time series data mining," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 164–181, 2011.

[2] V. Vo, J. Luo, and B. Vo, "Time series trend analysis based on k-means and support vector machine," *Comput. Inform.*, vol. 35, no. 1, pp. 111–127, 2016.

[3] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[5] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.

[6] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1604–1613, Dec. 2014.

[7] D. A. Keim, T. Munzner, F. Rossi, and M. Verleysen, "Bridging information visualization with machine learning (Dagstuhl seminar 15101)," Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Wadern, Germany, Dagstuhl Rep. 3, 2015, vol. 5, no. 3. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2015/5266

[8] G. A. Tsihrintzis, M. Virvou, E. Sakkopoulos, and L. C. Jain, "Applications of learning and analytics in intelligent systems," in *Machine Learning Paradigms*. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-15628-2_1.

[9] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "What you see is what you can change: Human-centered machine learning by interactive visualization," *Neurocomputing*, vol. 268, pp. 164–175, 2017.

[10] D. A. Keim, F. Rossi, T. Seidl, M. Verleysen, and S. Wrobel, "Information visualization, visual data mining and machine learning (Dagstuhl seminar 12081)," Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Wadern, Germany, Dagstuhl Rep. 2, 2012, vol. 2, no. 2. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2012/3506

[11] T. W. Liao, "Clustering of time series data—A survey," *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, 2005.

[12] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering—A decade review," *Inf. Syst.*, vol. 53, pp. 16–38, Oct./Nov. 2015, doi: 10.1016/j.is.2015.04.007.

[13] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 1, pp. 40–48, Jun. 2010.

[14] H. Yahyaoui and A. Al-Mutairi, "A feature-based trust sequence classification algorithm," *Inf. Sci.*, vol. 328, pp. 455–484, 2016.

[15] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of Time-Oriented Data*. London, U.K.: Springer, 2011, doi: 10.1007/978-0-85729-079-3.

[16] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale, "A review of temporal data visualizations based on space-time cube operations," in *Proc. Eurograph. Conf. Vis.*, 2014, doi: 10.2312/eurovisstar.20141171.

[17] S. Ko, I. Cho, S. Afzal, C. Yau, J. Chae, A. Malik, K. Beck, Y. Jang, W. Ribarsky, and D. S. Ebert, "A survey on visual analysis approaches for financial data," *Comput. Graph. Forum*, vol. 35, no. 3, pp. 599–617, 2016.

[18] A. Holzinger, "Introduction to machine learning and knowledge extraction (make)," *Mach. Learn. Knowl. Extr*, vol. 1, no. 1, pp. 1–20, 2017.

[19] C. Turkay, R. Laramee, and A. Holzinger, "On the challenges and opportunities in visualization for machine learning and knowledge extraction: A research agenda," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction*. Cham, Switzerland: Springer, 2017, pp. 191–198, doi: 10.1007/978-3-319-66808-6_13.

[20] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, *Mastering the Information Age Solving Problems With Visual Analytics*. Goslar, Germany: Eurographics Association, 2010.

[21] J. Montes, A. Sánchez, J. J. Valdés, M. S. Pérez, and P. Herrero, "Finding order in chaos: A behavior model of the whole grid," *Concurrency Comput., Pract. Exper.*, vol. 22, no. 11, pp. 1386–1415, 2010.

[22] D. Sacha, M. Kraus, D. A. Keim, and M. Chen, "VIS4ML: An ontology for visual analytics assisted machine learning," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 385–395, Jan. 2019.

[23] M. Rasmussen and G. Karypis, "gCLUTO: An interactive clustering, visualization, and analysis system," Dept. Comput. Sci. Eng., Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. UMN-CS TR-04, 2004, vol. 21, no. 7.

[24] G. Trajcevski, D. Gunopulos, C. C. Aggarwal, and C. Reddy, "Time-series data clustering," in *Data Clustering: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2013, pp. 357–375.

[25] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 3026–3037, Dec. 2014.

[26] M. Tucci and M. Raugi, "Analysis of spectral clustering algorithms for linear and nonlinear time series," in *Proc. 11th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2011, pp. 925–930.

[27] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual analytics: Scope and challenges," in *Visual Data Mining*. Berlin, Germany: Springer, 2008, pp. 76–90, doi: 10.1007/978-3-540-71080-6_6.

[28] A. Nielsen, *Practical Time Series Analysis: Prediction With Statistics and Machine Learning*. Newton, MA, USA: O'Reilly Media, 2019.

[29] E. A. Maharaj, P. D'Urso, and J. Caiado, *Time Series Clustering and Classification*. Boca Raton, FL, USA: CRC Press, 2019.

[30] I. Hotz and R. Peikert, "Definition of a multifield," in *Scientific Visualization*. Springer, 2014, pp. 105–109.

[31] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *Proc. Int. Conf. Web-Age Inf. Manage.* Cham, Switzerland: Springer, 2014, pp. 298–310, doi: 10.1007/978-3-319-08010-9_33.

[32] E. Lughofer and M. Sayed-Mouchaweh, *Predictive Maintenance in Dynamic Systems: Advanced Methods, Decision Support Tools and Real-World Applications*. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-05645-2.

[33] E. Lughofer, A.-C. Zavoianu, R. Pollak, M. Pratama, P. Meyer-Heye, H. Zörrer, C. Eitzinger, and T. Radauer, "Autonomous supervision and optimization of product quality in a multi-stage manufacturing process based on self-adaptive prediction models," *J. Process Control*, vol. 76, pp. 27–45, Apr. 2019, doi: 10.1016/j.jprocont.2019.02.005.

[34] F. Battke, S. Symons, and K. Nieselt, "Mayday-integrative analytics for expression data," *BMC Bioinf.*, vol. 11, no. 1, p. 121, 2010.

[35] M. Bagdouri, F. Chelaru, and Y. Wang, "Epiviz: A visual analytic tool for epigenetic features," Dept. Comput. Sci., Univ. Maryland, College Park, MD, USA, Tech. Rep., 2012. [Online]. Available: https://www.semanticscholar.org/paper/EpiViz-%3A-A-Visual-Analytic-Tool-for-Epigenetic-Bagdouri-Chelaru/b0b08ed0cc18525afb0bcc959b9a6535b847d67e

[36] D. H. Jeong, A. Darvish, K. Najarian, J. Yang, and W. Ribarsky, "Interactive visual analysis of time-series microarray data," *Vis. Comput.*, vol. 24, no. 12, pp. 1053–1066, 2008.

[37] A. Vogogias, J. Kennedy, and D. Archambault, "Hierarchical clustering with multiple-height branch-cut applied to short time-series gene expression data," in *EuroVis 2016—Posters*. The Eurographics Association, 2016, doi: 10.2312/eurp.20161127.

[38] M. Cho, B. Kim, H.-J. Bae, and J. Seo, "Stroscope: Multi-scale visualization of irregularly measured time-series data," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 5, pp. 808–821, May 2014.

[39] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Kern, and A. Sudjianto, "WireVis: Visualization of categorical, time-varying data from financial transactions," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct./Nov. 2007, pp. 155–162.

[40] C. Xie, W. Chen, X. Huang, Y. Hu, S. Barlowe, and J. Yang, "VAET: A visual analytics approach for E-transactions time-series," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1743–1752, Nov. 2014.

[41] C. Turkay, E. Kaya, S. Balcisoy, and H. Hauser, "Designing progressive and interactive analytics processes for high-dimensional data analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 131–140, Jan. 2017.

[42] S. T. Lei and K. Zhang, "A visual analytics system for financial time-series data," in *Proc. 3rd Int. Symp. Vis. Inf. Commun.*, 2010, p. 20.

[43] H. Ziegler, M. Jenny, T. Gruse, and D. A. Keim, "Visual market sector analysis for financial time series data," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 83–90.

[44] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer, "Visual cluster analysis of trajectory data with interactive Kohonen maps," *Inf. Vis.*, vol. 8, no. 1, pp. 14–29, 2009.

[45] T. Schreck, T. Tekušová, J. Kohlhammer, and D. Fellner, "Trajectory-based visual analysis of large financial time series data," *ACM SIGKDD Explor. Newslett.*, vol. 9, no. 2, pp. 30–37, 2007.

[46] A. Perer and J. Sun, "MatrixFlow: Temporal network visual analytics to track symptom evolution during disease progression," in *Proc. AMIA Annu. Symp.*, 2012, p. 716.

[47] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao, "EventThread: Visual summarization and stage analysis of event sequence data," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 56–65, Jan. 2018.

[48] Y. Wang, T. Wu, Z. Chen, Q. Luo, and H. Qu, "STAC: Enhancing stacked graphs for time series analysis," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Apr. 2016, pp. 234–238.

[49] L. Wilkinson, "Visualizing big data outliers through distributed aggregation," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 256–266, Jan. 2018.

[50] T. Lin, F. Guo, Y. Wu, B. Zhu, F. Zhang, H. Qu, and W. Chen, "Tievis: Visual analytics of evolution of interpersonal ties," in *Proc. Int. Conf. Technol. E-Learn. Digit. Entertainment*. Cham, Switzerland: Springer, 2016, pp. 412–424, doi: 10.1007/978-3-319-40259-8_36.

[51] A. Meidiana and S.-H. Hong, "Multistory: Visual analytics of dynamic multi-relational networks," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Apr. 2015, pp. 75–79.

[52] S. Hadlak, H. Schumann, C. H. Cap, and T. Wollenberg, "Supporting the visual analysis of dynamic networks by clustering associated temporal attributes," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2267–2276, Dec. 2013.

[53] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer, "Visual analysis of time-series similarities for anomaly detection in sensor networks," *Comput. Graph. Forum*, vol. 33, no. 3, pp. 401–410, 2014.

[54] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Reducing snapshots to points: A visual analytics approach to dynamic network exploration," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 1–10, Aug. 2016.

[55] F. Zhou, W. Huang, Y. Zhao, Y. Shi, X. Liang, and X. Fan, "ENTVis: A visual analytic tool for entropy-based network traffic anomaly detection," *IEEE Comput. Graph. Appl.*, vol. 35, no. 6, pp. 42–50, Nov./Dec. 2015.

[56] T. Fujiwara, J. K. Li, M. Mubarak, C. Ross, C. D. Carothers, R. B. Ross, and K.-L. Ma, "A visual analytics system for optimizing the performance of large-scale networks in supercomputing systems," *Vis. Inform.*, vol. 2, no. 1, pp. 98–110, 2018.

[57] N. Cao, C. Shi, S. Lin, J. Lu, Y.-R. Lin, and C.-Y. Lin, "TargetVue: Visual analysis of anomalous user behaviors in online communication systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 280–289, Jan. 2016.

[58] M. C. Hao, M. Marwah, H. Janetzko, D.-I. A. Keim, U. Dayal, R. Sharma, D. Patnaik, and N. Ramakrishnan, "Visual analysis of frequent patterns in large time series," in *Proc. IEEE Symp. Visual Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 227–228.

[59] C. Muelder, B. Zhu, W. Chen, H. Zhang, and K.-L. Ma, "Visual analysis of cloud computing performance using behavioral lines," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 6, pp. 1694–1704, Feb. 2016.

[60] G. Sharma, G. Shroff, A. Pandey, B. Singh, G. Sehgal, K. Paneri, and P. Agarwal, "Multi-sensor visual analytics supported by machine-learning models," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 668–674.

[61] L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, and Z. Su, "SAVE: Sensor anomaly visualization engine," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2011, pp. 201–210.

[62] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer, "Visplause: Visual data quality assessment of many time series using plausibility checks," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 641–650, Jan. 2017.

[63] Y. Chen, P. Xu, and L. Ren, "Sequence synopsis: Optimize visual summary of temporal event data," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 45–55, Sep. 2018.

[64] G. Andrienko, N. Andrienko, and S. Wrobel, "Visual analytics tools for analysis of movement data," *ACM SIGKDD Explor. Newslett.*, vol. 9, no. 2, pp. 38–46, 2007.

[65] G. Andrienko and N. Andrienko, "Spatio-temporal aggregation for visual analysis of movements," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2008, pp. 51–58.

[66] G. Andrienko, N. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel, "From movement tracks through events to places: Extracting and characterizing significant places from mobility data," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2011, pp. 161–170.

[67] G. Andrienko, N. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel, "Scalable analysis of movement data for extracting and exploring significant places," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 7, pp. 1078–1094, Jul. 2013.

[68] M. Lu, Z. Wang, and X. Yuan, "TrajRank: Exploring travel behaviour on a route by trajectory ranking," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Apr. 2015, pp. 311–318.

[69] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti, "Interactive visual clustering of large collections of trajectories," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2009, pp. 3–10.

[70] M. Riveiro, M. Lebram, and M. Elmer, "Anomaly detection for road traffic: A visual analytics framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2260–2270, Aug. 2017.

[71] I. Kalamaras, A. Zamichos, A. Salamanis, A. Drosou, D. D. Kehagias, G. Margaritis, S. Papadopoulos, and D. Tzovaras, "An interactive visual analytics platform for smart intelligent transportation systems management," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 487–496, Feb. 2018.

[72] G. Andrienko, N. Andrienko, M. Mladenov, M. Mock, and C. Pölitz, "Identifying place histories from activity traces with an eye to parameter impact," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 5, pp. 675–688, May 2012.

[73] G. L. Andrienko, N. V. Andrienko, G. Fuchs, A.-M. O. Raimond, J. Symanzik, and C. Ziemlicki, "Extracting semantics of individual places from movement data by analyzing temporal patterns of visits," in *Proc. COMP SIGSPATIAL*, 2013, pp. 9–15.

[74] J. Chae, G. Wang, B. Ahlbrand, M. B. Gorantla, J. Zhang, S. Chen, H. Xu, J. Zhao, W. Hatton, A. Malik, S. Ko, and D. S. Ebert, "Visual analytics of heterogeneous data for criminal event analysis vast challenge 2015: Grand challenge," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2015, pp. 149–150.

[75] J. Pu, S. Liu, P. Xu, H. Qu, and L. M. Ni, "MViewer: Mobile phone spatiotemporal data viewer," *Frontiers Comput. Sci.*, vol. 8, no. 2, pp. 298–315, 2014.

[76] Z. Shen and K.-L. Ma, "MobiVis: A visualization system for exploring mobile data," in *Proc. IEEE Pacific Vis. Symp. (PacificVIS)*, Mar. 2008, pp. 175–182.

[77] J. Zhao, G. Wang, J. Chae, H. Xu, S. Chen, W. Hatton, S. Towers, M. B. Gorantla, B. Ahlbrand, J. Zhang, A. Malik, S. Ko, and D. S. Ebert, "ParkAnalyzer: Characterizing the movement patterns of visitors VAST 2015 mini-challenge 1," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2015, pp. 179–180.

[78] T. von Landesberger, F. Brodkorb, P. Roskosch, N. Andrienko, G. Andrienko, and A. Kerren, "MobilityGraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 11–20, Jan. 2016.

[79] A. Biswas, G. Lin, X. Liu, and H.-W. Shen, "Visualization of time-varying weather ensembles across multiple resolutions," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 841–850, Jan. 2017.

[80] H. Senaratne, M. Mueller, M. Behrisch, F. Lalanne, J. Bustos-Jiménez, J. Schneidewind, D. Keim, and T. Schreck, "Urban mobility analysis with mobile network data: A visual analytics approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1537–1546, Aug. 2018.

[81] L. Stopar, P. Skraba, M. Grobelnik, and D. Mladenic, "Streamstory: Exploring multivariate time series on multiple scales," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 4, pp. 1788–1802, Apr. 2018.

[82] L. Gao, H. A. Campbell, O. R. Bidder, and J. Hunter, "A Web-based semantic tagging and activity recognition system for species' accelerometry data," *Ecological Informat.*, vol. 13, pp. 47–56, Jan. 2013, doi: 10.1016/j.ecoinf.2012.09.003.

[83] J. S. Walker, M. W. Jones, R. S. Laramee, O. R. Bidder, H. J. Williams, R. Scott, E. L. Shepard, and R. P. Wilson, "TimeClassifier: A visual

[84] J. Bernard, N. Wilhelm, M. Scherer, T. May, and T. Schreck, "TimeSeriesPaths: Projection-based explorative analysis of multivariate time series data," *J. WSCG*, pp. 97–106, 2012. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:352-227012

[85] R. Kincaid and H. Lam, "Line graph explorer: Scalable display of line graphs using focus+ context," in *Proc. Work. Conf. Adv. Vis. Interfaces*, 2006, pp. 404–411.

[86] J. Li, Z. Xiao, H.-Q. Zhao, Z.-P. Meng, and K. Zhang, "Visual analytics of smogs in China," *J. Vis.*, vol. 19, no. 3, pp. 461–474, 2016.

[87] J. Li, K. Zhang, and Z.-P. Meng, "Vismate: Interactive visual analysis of station-based observation data on climate changes," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 133–142.

[88] S. Martin and T.-T. Quach, "Interactive visualization of multivariate time series data," in *Proc. Int. Conf. Augmented Cognition*. Cham, Switzerland: Springer, 2016, pp. 322–332, doi: 10.1007/978-3-319-39952-2_31.

[89] Q. Shu, H. Guo, J. Liang, L. Che, J. Liu, and X. Yuan, "EnsembleGraph: Interactive visual analysis of spatiotemporal behaviors in ensemble simulation data," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Apr. 2016, pp. 56–63.

[90] W. Wu, Y. Zheng, H. Qu, W. Chen, E. Gröller, and L. M. Ni, "BoundarySeer: Visual analysis of 2D boundary changes," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 143–152.

[91] J. Bernard, N. Wilhelm, B. Krüger, T. May, T. Schreck, and J. Kohlhammer, "MotionExplorer: Exploratory search in human motion capture data based on hierarchical aggregation," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2257–2266, Dec. 2013.

[92] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl, "VA²: A visual analytics approach for evaluating visual analytics applications," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 61–70, Jan. 2016.

[93] O. Purwantiningsih, A. Sallaberry, S. Andary, A. Seilles, and J. Azé, "Visual analysis of body movement in serious games for healthcare," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Apr. 2016, pp. 229–233.

[94] J. G. Kim, M. Snodgrass, M. Pietrowicz, and K. Karahalios, "Visual analysis of relationships between behavioral and physiological sensor data," in *Proc. Int. Conf. Healthcare Inform. (ICHI)*, Oct. 2015, pp. 170–179.

[95] C. Turkay, J. Parulek, N. Reuter, and H. Hauser, "Interactive visual analysis of temporal cluster structures," *Comput. Graph. Forum*, vol. 30, no. 3, pp. 711–720, Jun. 2011.

[96] A. Soriano-Vargas, B. C. Vani, M. H. Shimabukuro, J. F. Monico, M. C. F. Oliveira, and B. Hamann, "Visual analytics of time-varying multivariate ionospheric scintillation data," *Comput. Graph.*, vol. 68, pp. 96–107, Nov. 2017, doi: 10.1016/j.cag.2017.08.013.

[97] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," 2014, *arXiv:1403.2877*. [Online]. Available: https://arxiv.org/abs/1403.2877

[98] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining Knowl. Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[99] K. Tornai, L. Kovács, A. Oláh, R. Drenyovszki, I. Pintér, D. Tisza, and J. Levendovszky, "Classification for consumption data in smart grid based on forecasting time series," *Electr. Power Syst. Res.*, vol. 141, pp. 191–201, Dec. 2016, doi: 10.1016/j.epsr.2016.07.018.

[100] L. Wei and E. Keogh, "Semi-supervised time series classification," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 748–753.

[101] J. Ye, C. Xiao, R. M. Esteves, and C. Rong, "Time series similarity evaluation based on spearman's correlation coefficients and distance measures," in *Proc. Int. Conf. Cloud Comput. Big Data Asia*. Cham, Switzerland: Springer, 2015, pp. 319–331, doi: 10.1007/978-3-319-28430-9_24.

[102] P. Buono, A. Aris, C. Plaisant, A. Khella, and B. Shneiderman, "Interactive pattern search in time series," in *Visualization and Data Analysis 2005*, vol. 5669, R. F. Erbacher, J. C. Roberts, M. T. Grohn, and K. Borner, Eds. Bellingham, WA, USA: SPIE, 2005, pp. 175–186, doi: 10.1117/12.587537.

[103] D. Yu, X. Yu, Q. Hu, J. Liu, and A. Wu, "Dynamic time warping constraint learning for large margin nearest neighbor classification," *Inf. Sci.*, vol. 181, no. 13, pp. 2787–2796, 2011.

[104] J. Zhao and L. Itti, "ShapeDTW: Shape dynamic time warping," 2016, *arXiv:1606.01601*. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320317303710

[105] C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *Proc. SIAM*, 2004, pp. 11–22, doi: 10.1137/1.9781611972740.2.

[106] A. Mueen and E. Keogh, "Extracting optimal performance from dynamic time warping," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 2129–2130.

[107] M. Kotas, J. M. Leski, and T. Moroń, "Dynamic time warping based on modified alignment costs for evoked potentials averaging," in *Man–Machine Interactions 4*. Cham, Switzerland: Springer, 2016, pp. 305–314, doi: 10.1007/978-3-319-23437-3_26.

[108] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *SIGMOD Rec.*, vol. 23, no. 2, pp. 419–429, 1994, doi: 10.1145/191843.191925.

[109] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, Seattle, WA, USA, 1994, vol. 10, no. 16, pp. 359–370.

[110] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, 2005.

[111] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. 18th Int. Conf. Data Eng.*, Feb./Mar. 2002, pp. 673–684.

[112] S. Smith, *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Oxford, U.K.: Newnes, 2013.

[113] Y. Lu, R. Krüger, D. Thom, F. Wang, S. Koch, T. Ertl, and R. Maciejewski, "Integrating predictive analytics and social media," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 193–202.

[114] D. Sacha, M. Kraus, J. Bernard, M. Behrisch, T. Schreck, Y. Asano, and D. A. Keim, "SOMFlow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 120–130, Jan. 2018.

[115] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu, "Analyzing the training processes of deep generative models," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 77–87, Jan. 2018.

[116] H. Kaya and Ş. Gündüz-Öğüdücü, "A distance based time series classification framework," *Inf. Syst.*, vol. 51, pp. 27–42, Jul. 2015, doi: 10.1016/j.is.2015.02.005.

[117] K. Yang and C. Shahabi, "A PCA-based similarity measure for multivariate time series," in *Proc. 2nd ACM Int. Workshop Multimedia Databases*, 2004, pp. 65–74.

[118] A. Singhal and D. E. Seborg, "Clustering multivariate time-series data," *J. Chemometrics*, vol. 19, no. 8, pp. 427–438, 2005.

[119] K. Yang and C. Shahabi, "On the stationarity of multivariate time series for correlation-based data analysis," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, p. 4.

[120] R. H. Lesch, Y. Caillé, and D. Lowe, "Component analysis in financial time series," in *Proc. IEEE/IAFE Conf. Comput. Intell. Financial Eng. (CIFEr)*, Apr. 1999, pp. 183–190.

[121] G. Dong and J. Pei, *Sequence Data Mining*, vol. 33. Boston, MA, USA: Springer, 2007, doi: 10.1007/978-0-387-69937-0.

[122] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," in *Proc. Int. Conf. Found. Data Org. Algorithms*. Berlin, Germany: Springer, 1993, pp. 69–84, doi: 10.1007/3-540-57301-1_5.

[123] D. Rafiei and A. O. Mendelzon, "Querying time series data based on similarity," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 5, pp. 675–693, Sep./Oct. 2000.

[124] G. J. Janacek, A. J. Bagnall, and M. Powell, "A likelihood ratio distance measure for the similarity between the Fourier transform of time series," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2005, pp. 737–743, doi: 10.1007/11430919_85.

[125] I. Popivanov and R. J. Miller, "Similarity search over time-series data using wavelets," in *Proc. 18th Int. Conf. Data Eng.*, Feb./Mar. 2002, pp. 212–221.

[126] K.-P. Chan and A. W.-C. Fu, "Efficient time series matching by wavelets," in *Proc. 15th Int. Conf. Data Eng.*, Mar. 1999, pp. 126–133.

[127] C. C. Aggarwal, "On effective classification of strings with wavelets," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 163–172.

[128] D. Li, T. F. D. A. Bissyande, J. Klein, and Y. Le Traon, "Time series classification with discrete wavelet transformed data: Insights from an empirical study," in *Proc. 28th Int. Conf. Softw. Eng. Knowl. Eng. (SEKE)*, 2016, pp. 1361–1377, doi: 10.1142/S0218194016400088.

[129] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 947–956.

[130] Z. Xing, J. Pei, S. Y. Philip, and K. Wang, "Extracting interpretable features for early classification on time series," in *Proc. SDM*, vol. 11. Philadelphia, PA, USA: SIAM, 2011, pp. 247–258.

[131] M. Ali, M. W. Jones, X. Xie, and M. Williams, "TimeCluster: Dimension reduction applied to temporal data for visual analytics," *Vis. Comput.*, vol. 35, no. 6, pp. 1013–1026, 2019.

[132] X. Xie, X. Cai, J. Zhou, N. Cao, and Y. Wu, "A semantic-based method for visualizing large image collections," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 7, pp. 2362–2377, Jul. 2019.

[133] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. Hoboken, NJ, USA: Wiley, 2009.

[134] J. MacQueen, "Some methods for classification and analysis of multivariate observa-tions," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, 1967, vol. 1, no. 14, pp. 281–297.

[135] L. Rokach and O. Maimon, "Clustering methods," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2005, pp. 321–352, doi: 10.1007/0-387-25465-X_15.

[136] C. Goutte, P. Toft, E. Rostrup, F. Å Nielsen, and L. K. Hansen, "On clustering fMRI time series," *NeuroImage*, vol. 9, no. 3, pp. 298–310, 1999.

[137] V. Niennattrakul and C. A. Ratanamahatana, "On clustering multimedia time series data using k-means and dynamic time warping," in *Proc. Int. Conf. Multimedia Ubiquitous Eng. (MUE)*, Apr. 2007, pp. 733–738.

[138] F. Klawonn, "Fuzzy clustering: Insights and a new approach," *Mathware Soft Comput.*, vol. 11, no. 3, pp. 125–142, 2004. [Online]. Available: http://hdl.handle.net/2099/3642

[139] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973, doi: 10.1080/01969727308546046.

[140] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York, NY, USA: Springer, 2013, doi: 10.1007/978-1-4757-0450-1.

[141] S. Eschrich, J. Ke, L. O. Hall, and D. B. Goldgof, "Fast accurate fuzzy clustering through data reduction," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 2, pp. 262–270, Apr. 2003.

[142] C. S. Möller-Levet, F. Klawonn, K.-H. Cho, and O. Wolkenhauer, "Fuzzy clustering of short time-series and unevenly distributed sam-pling points," in *Proc. Int. Symp. Intell. Data Anal.* Berlin, Germany: Springer, 2003, pp. 330–340, doi: 10.1007/978-3-540-45231-7_31.

[143] J. C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 1, pp. 1–8, Jan. 1980.

[144] R. L. Cannon, J. V. Dave, and J. C. Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 2, pp. 248–255, Mar. 1986.

[145] R. Krishnapuram, A. Joshi, O. Nasraoui, and L. Yi, "Low-complexity fuzzy relational clustering algorithms for Web mining," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 595–607, Aug. 2001.

[146] W. Meesrikamolkul, V. Niennattrakul, and C. A. Ratanamahatana, "Shape-based clustering for time series data," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2012, pp. 530–541, doi: 10.1007/978-3-642-30217-6_44.

[147] V. Hautamaki, P. Nykanen, and P. Franti, "Time-series clustering by approximate prototypes," in *Proc. 19th Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2008, pp. 1–4.

[148] C. Guo, H. Jia, and N. Zhang, "Time series clustering based on ICA for stock data analysis," in *Proc. 4th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Oct. 2008, pp. 1–4.

[149] K. Kalpakis, D. Gada, and V. Puttagunta, "Distance measures for effective clustering of ARIMA time-series," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov./Dec. 2001, pp. 273–280.

[150] X. Golay, S. Kollias, G. Stoll, D.-E. Meier, A. Valavanis, and P. Boesiger, "A new correlation-based fuzzy logic clustering algorithm for fMRI," *Magn. Reson. Med.*, vol. 40, no. 2, pp. 249–260, 1998.

[151] P. D'Urso, C. Cappelli, D. D. Lallo, and R. Massari, "Clustering of financial time series," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 9, pp. 2114–2129, 2013.

[152] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 218–237, Jan. 2008.

[153] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: Implications for previous and future research," *Knowl. Inf. Syst.*, vol. 8, no. 2, pp. 154–177, 2005.

[154] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, nos. 2–3, pp. 107–145, 2001.

[155] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, 2003, pp. 2–11.

[156] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," *ACM SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, 1998.

[157] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, pp. 103–114, Jun. 1996.

[158] S. Aghabozorgi, M. R. Saybani, and T. Y. Wah, "Incremental clustering of time-series by fuzzy clustering," *J. Inf. Sci. Eng.*, vol. 28, no. 4, pp. 671–688, 2012.

[159] J. J. Van Wijk and E. R. Van Selow, "Cluster and calendar based visualization of time series data," in *Proc. IEEE Symp. Inf. Vis. (InfoVis)*, Oct. 1999, pp. 4–9.

[160] M. Simonsen, T. Mailund, and C. N. S. Pedersen, *Rapid Neighbour-Joining*. Berlin, Germany: Springer, 2008, pp. 113–122, doi: 10.1007/978-3-540-87361-7_10.

[161] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[162] M. Shanmuganathan, "An overview of IRFS on Canadian GAAP—Self-organizing maps (SOMs)^{MS}," in *Proc. SAI Comput. Conf. (SAI)*, Jul. 2016, pp. 100–107.

[163] V. J. Lobo, "Application of self-organizing maps to the maritime environment," in *Proc. Inf. Fusion Geograph. Inf. Syst.* Berlin, Germany: Springer, 2009, pp. 19–36, doi: 10.1007/978-3-642-00304-2_2.

[164] F. Bação, V. Lobo, and M. Painho, "The self-organizing map, the geo-SOM, and relevant variants for geosciences," *Comput. Geosci.*, vol. 31, no. 2, pp. 155–163, 2005.

[165] M. Chang, H.-J. Yu, and J.-S. Heh, "Evolutionary self-organizing map," in *Proc. IEEE Int. Joint Conf. Neural Netw. IEEE World Congr. Comput. Intell.*, vol. 1, May 1998, pp. 680–685.

[166] T. Kohonen, "The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection," in *Proc. ICANN*, vol. 95, 1995, pp. 3–10.

[167] T. Kohonen, "Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map," *Biological*, vol. 75, no. 4, pp. 281–291, 1996.

[168] J. Walter and H. Ritter, "Rapid learning with parametrized self-organizing maps," *Neurocomputing*, vol. 12, no. 2, pp. 131–153, 1996.

[169] H. Yin, "Visualisation induced SOM (VISOM)," in *Advances in Selforganising Maps*. London, U.K.: Springer, 2001, pp. 81–88, doi: 10.1007/978-1-4471-0715-6_12.

[170] H. Yin, "Visom-a novel method for multivariate data projection and structure visualization," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 237–243, Aug. 2002.

[171] H. Yin and N. M. Allinson, "Self-organizing mixture networks for probability density estimation," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 405–411, Mar. 2001.

[172] H. Yin, "The self-organizing maps: Background, theories, extensions and applications," in *Computational Intelligence: A Compendium*. Berlin, Germany: Springer, 2008, pp. 715–762, doi: 10.1007/978-3-540-78293-3_17.

[173] X. Wang, K. A. Smith, R. Hyndman, and D. Alahakoon, "A scalable method for time series clustering," in *Proc. Unrefereed Res. Papers*, vol. 1, 2004. [Online]. Available: http://robjhyndman.com/papers/wang.pdf

[174] M. Varstal, J. D. R. Millán, and J. Heikkonen, "A recurrent self-organizing map for temporal sequence processing," in *Proc. Int. Conf. Artif. Neural Netw.* Berlin, Germany: Springer, 1997, pp. 421–426, doi: 10.1007/BFb0020191.

[175] T. Voegtlin, "Recursive self-organizing maps," *Neural Netw.*, vol. 15, no. 8, pp. 979–991, 2002.

[176] T.-C. Fu, F.-L. Chung, V. Ng, and R. Luk, "Pattern discovery from stock time series using self-organizing maps," in *Proc. Workshop Notes KDD Workshop Temporal Data Mining*, 2001, pp. 26–29.

[177] G. Sheikholeslami, S. Chatterjee, and A.-D. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases," in *Proc. VLDB*, vol. 98, 1998, pp. 428–439.

[178] W. Wang, J. Yang, and R. Muntz, "Sting: A statistical information grid approach to spatial data min-ing," in *Proc. VLDB*, vol. 97, 1997, pp. 186–195.

[179] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.

[180] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, 1999.

[181] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[182] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based GMM," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 145–154.

[183] P. Esling and A. Carlos, "Time-series data mining," *ACM Comput. Surv.*, vol. 45, no. 1, p. 12, 2012.

[184] G. E. Batista, X. Wang, and E. J. Keogh, "A complexity-invariant distance measure for time series," in *Proc. SDM*, vol. 11. Philadelphia, PA, USA: SIAM, 2011, pp. 699–710.

[185] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[186] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 1033–1040.

[187] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 262–270.

[188] B. De Ville and P. Neville, *Decision Trees for Analytics Using SAS Enterprise Miner*. Cary, NC, USA: SAS Institute, 2013.

[189] P.-N. Tan, M. Steinbach, and V. Kumar, "Classification: Basic concepts, decision trees, and model evaluation," in *Introduction to Data Mining*, vol. 1. Boston, MA, USA: Addison-Wesley, 2006, pp. 145–205.

[190] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2014.

[191] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[192] D. Steinberg and P. Colla, "Cart: Classification and regression trees," *Top Ten Algorithms Data Mining*, vol. 9, p. 179, Apr. 2009, doi: 10.1201/9781420089653.

[193] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, and S. Y. Philip, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[194] G. Ritschard, "Chaid and earlier supervised tree methods," Institut d'Economie et Econométrie, Univ. Genève, Tech. Rep., 2010, doi: 10.4324/9780203403020.

[195] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers—A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 476–487, Nov. 2005.

[196] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[197] K. Ellis, J. Kerr, S. Godbole, G. Lanckriet, D. Wing, and S. Marshall, "A random forest classifier for the prediction of energy expenditure and type of physical activity from wrist and hip accelerometers," *Physiolog. Meas.*, vol. 35, no. 11, p. 2191, 2014.

[198] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1. Berlin, Germany: Springer, 2001.

[199] M. Deshpande and G. Karypis, "Evaluation of techniques for classifying biological sequences," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2002, pp. 417–431, doi: 10.1007/3-540-47887-6_41.

[200] C. S. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification," in *Proc. Pacific Symp. Biocomput.*, 2002, vol. 7, no. 7, pp. 566–575.

[201] S. Sonnenburg, G. Rätsch, and C. Schäfer, "Learning interpretable SVMs for biological sequence classification," in *Proc. Annu. Int. Conf. Res. Comput. Mol. Biol.* Berlin, Germany: Springer, 2005, pp. 389–407, doi: 10.1007/11415770_30.

[202] M. Li and R. Sleep, "A robust approach to sequence classification," in *Proc. 17th IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2005, p. 5.

[203] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, Aug. 2004, pp. 32–36.

[204] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[205] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.

[206] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.

[207] S. Sonnenburg, G. Rätsch, and B. Schölkopf, "Large scale genomic sequence svm classifiers," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 848–855.

[208] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *J. Mach. Learn. Res.*, vol. 2, pp. 419–444, Feb. 2002.

[209] S. K. Chalup and A. Mitschele, "Kernel methods in finance," in *Handbook on Information Technology in Finance*. Berlin, Germany: Springer, 2008, pp. 655–687, doi: 10.1007/978-3-540-49487-4_27.

[210] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Jul. 2006.

[211] A. Nanopoulos, R. Alcock, and Y. Manolopoulos, "Feature-based classification of time-series data," *Int. J. Comput. Res.*, vol. 10, no. 3, pp. 49–61, 2001.

[212] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI)*, Buenos Aires, Argentina, 2015, pp. 25–31.

[213] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services (MobiCASE)*, Nov. 2014, pp. 197–205.

[214] M. Hüsken and P. Stagge, "Recurrent neural networks for time series classification," *Neurocomputing*, vol. 50, pp. 223–235, Jan. 2003, doi: 10.1016/S0925-2312(01)00706-8.

[215] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," 2016, *arXiv:1603.06995*. [Online]. Available: https://arxiv.org/abs/1603.06995

[216] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 96–112, 2016.

[217] T.-M. Rhyne, "Does the difference between information and scientific visualization really matter?" *IEEE Comput. Graph. Appl.*, vol. 23, no. 3, pp. 6–8, May 2003.

[218] D. A. Keim, "Information visualization and visual data mining," *IEEE Trans. Vis. Comput. Graphics*, vol. 8, no. 1, pp. 1–8, Jan./Mar. 2002.

[219] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister, "What makes a visualization memorable?" *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2306–2315, Oct. 2013.

[220] F. Hartwig and B. E. Dearing, *Exploratory Data Analysis*. Newbury Park, CA, USA: Sage, 1979.

[221] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proc. IEEE Symp. Vis. Lang.*, Sep. 1996, pp. 336–343.

[222] L. Wilkinson, *The Grammar of Graphics*. New York, NY, USA: Springer, 2005, doi: 10.1007/0-387-28695-0.

[223] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*. San Mateo, CA, USA: Morgan Kaufmann, 1999.

[224] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," in *Proc. Int. Conf. Intell. Anal.*, vol. 5, 2005, pp. 2–4.

[225] Z. Liu and J. Stasko, "Mental models, visual reasoning and interaction in information visualization: A top-down perspective," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 999–1008, Nov./Dec. 2010.

[226] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2376–2385, Oct. 2013.

[227] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL, USA: CRC Press, 2014.

[228] B. Spencer, M. W. Jones, and I. S. Lim, "A visualization tool used to develop new photon mapping techniques," *Comput. Graph. Forum*, vol. 34, no. 1, pp. 127–140, 2015.

[229] E. Lughofer, R. Richter, U. Neissl, W. Heidl, C. Eitzinger, and T. Radauer, "Explaining classifier decisions linguistically for stimulating and improving operators labeling behavior," *Inf. Sci.*, vol. 420, pp. 16–36, 2017.

[230] M. Robnik-Šikonja and I. Kononenko, "Explaining classifications for individual instances," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 589–600, 2008.

[231] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÃžller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, Jun. 2010.

[232] T. Munzner, "A nested model for visualization design and validation," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 6, pp. 921–928, Nov./Dec. 2009.

[233] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[234] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.

[235] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4829–4837.

[236] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5188–5196.

[237] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Univ. Montreal, Montreal, QC, Canada, Tech. Rep. 1341, 2009, p. 3.

[238] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034*. [Online]. Available: https://arxiv.org/abs/1312.6034

[239] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015, *arXiv:1506.06579*. [Online]. Available: https://arxiv.org/abs/1506.06579

[240] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 91–100, Jan. 2017.

[241] F.-Y. Tzeng and K.-L. Ma, "Opening the black box—Data driven visualization of neural networks," in *Proc. IEEE Vis. (VIS)*, Oct. 2005, pp. 383–390.
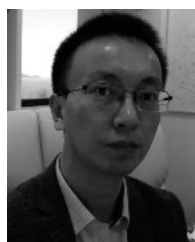
MOHAMMED ALI is currently pursuing the Ph.D. degree in computer science with Swansea University in the Visual Computing Research Group and the Computer Vision and Machine Learning Laboratory, under the supervision of Prof. Mark Jones. He is a Lecturer with the Department of Computer Science, King Khalid University, SA. His research interests include data mining and knowledge discovery, information visualization, visual analytics, and machine learning.

ALI ALQAHTANI is currently pursuing the Ph.D. degree in computer science with the Computer Vision and Machine Learning Laboratory, Swansea University, under the supervision of Prof. Xianghua Xie. He is a Lecturer with the Department of Computer Science, King Khalid University, SA. His research interests include deep learning, machine learning, and image processing and visualization.

MARK W. JONES received the B.Sc. and Ph.D. degrees from Swansea University. He is currently a Professor with the Department of Computer Science, Swansea University, where he leads the Visual Computing Research Group. His research interests include visualization, visual analytics, data science, machine learning, and global illumination and associated algorithms and data structures.

XIANGHUA XIE received the M.Sc. (Hons.) and Ph.D. degrees in computer science from the University of Bristol, Bristol, U.K., in 2002 and 2006, respectively. He is currently a Full Professor with the Department of Computer Science, Swansea University, Swansea, U.K. He was a recipient of an RCUK academic fellowship. He is also leading the Computer Vision and Machine Learning Laboratory, Swansea University. He has published more than 140 refereed conference and journal publications and co-edited several conference proceedings. His research interests include various aspects of pattern recognition, and machine intelligence and their applications to real-world problems. He is a member of BMVA. He was a recipient of an RCUK academic fellowship. He is an Associate Editor of a number of journals, including *IET Computer Vision*.

● ● ●