Human-in-the-Loop: Visual Analytics for Building Models Recognising Behavioural Patterns in Time Series

Natalia Andrienko, *Fraunhofer Institute IAIS, Sankt Augustin, Germany and Lamarr Institute, Germany and City, University of London, UK*

Gennady Andrienko, *Fraunhofer Institute IAIS, Sankt Augustin, Germany and Lamarr Institute, Germany and City, University of London, UK*

Alexander Artikis, *NCSR "Demokritos", Greece and University of Piraeus, Greece*

Periklis Mantenoglou, *National and Kapodistrian University of Athens, Greece and NCSR "Demokritos", Greece*

Salvatore Rinzivillo, *KDDLab, CNR-ISTI, Italy*

Abstract—Detecting complex behavioural patterns in temporal data, like moving object trajectories, often relies on precise formal specifications derived from vague domain concepts. However, such methods are sensitive to noise and minor fluctuations, leading to missed pattern occurrences. Conversely, machine learning (ML) approaches require abundant labeled examples, posing practical challenges. Our visual analytics approach enables domain experts to derive, test, and combine interval-based features to discriminate patterns and generate training data for ML algorithms. Visual aids enhance recognition and characterisation of expected patterns and discovery of unexpected ones. Case studies demonstrate feasibility and effectiveness of the approach, which offers a novel framework for integrating human expertise and analytical reasoning with ML techniques, advancing data analytics.

One of common tasks in analysing time-referenced data, such as multivariate time series and trajectories of moving objects, is to find time intervals where the manner, or pattern, of data variation is indicative of particular kinds of dynamic behaviour. Automatic detection of such patterns by means of computer algorithms requires precise specification of what values of attributes may occur and how the data are expected to vary. In many application domains, however, patterns of interest have no exact definitions. What can be elicited from domain experts is often far from being distinct and precise, for example, "A flock is a large enough group of objects moving close to each other for a certain time". Translation of such description to a form suitable for automated search involves introducing parameters and thresholds; see, for example, the formal definition of the flock pattern [1]: "Let $m, k \in N$, and let $r > 0$ be a constant. Consider a set of trajectories, where each trajectory consists of T line segments. A flock in a time interval $I = [t_i, t_j]$, where $j - i + 1 \geq k$, consists of at least m entities such that for every point in time within I there is a disk of radius r that contains all the m entities".

Formalisation of vague definitions elicited from domain experts entails two problems. First, the choice of appropriate parameter settings may not be obvious, while different choices may lead to very diverse results. Second, after the parameters are set, the definitions become rigid and intolerant to even minor data noise and small deviations from the thresholds. Imagine, for example, that just for a single time moment one of the m entities moving in a flock steps out from the disk of radius r. This breaks the time interval I in which the conditions of the formal definition of a flock hold. If the lengths of the sub-intervals are less than k, the flock will not be detected.

We encountered the problem of definition rigidity in exploring the work of a knowledge-based system designed to detect complex activity patterns in vessel

Digital Object Identifier 10.1109/XXX.0000.0000000

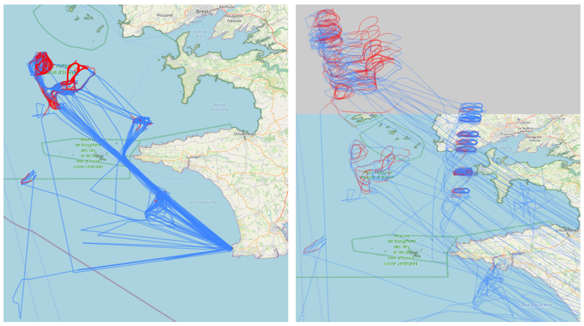

FIGURE 1. Trajectory of one selected fishing vessel is shown on a map (left) and in a space-time cube (right). Segments involved in automatically identified trawling patterns are painted in red, and the remaining segments in blue. It can be seen that most of the loop-shaped segments indicating trawling activities have not been recognised as trawling.

movement [2]. The system applies Event Calculus [3] to a set of formal definitions, many of which involve constant thresholds such as speed bounds, minimal change in movement direction, frequency of changes, and minimal duration of an activity. Upon observing that the system fails to recognise a significant number of visually identifiable pattern instances (as illustrated in Fig. 1 by example of one vessel trajectory), we employed interactive visualisation to investigate the data used for the inference. We found that the minimal activity duration was often not formally reached due to occasional breaks in the fulfillment of the rule conditions, which, in turn, happened because of data noise and small variations of attribute values around the thresholds.

Hence, formalisation of human-defined concepts may not be a good approach in tasks requiring the tolerance and flexibility of human reasoning. Probabilistic methods of pattern recognition (e.g., [4]) can be less sensitive to data noise, but they still assume that pattern specifications obtained from experts are complete and precise, which is not always the case.

Opposite to specification-driven approaches, machine learning methods strive to acquire the ability of pattern recognition by generalising from labelled data examples. Due to the generalisation, the resulting classification models can be sufficiently flexible regarding data variability. However, machine learning methods require an abundant supply of representative training examples, which may be very problematic. While domain experts can usually easily identify a pattern (or pattern absence) given an appropriately represented piece of data, their time is too costly to be spent for considering and labelling a large number of individually shown examples.

The inherent problems of the knowledge- and data-driven approaches call for hybrid solutions that would be able to effectively leverage expert knowledge while accommodating the flexibility of human reasoning [5], [6], abstractive perception and capability to give meaning to visual patterns [7].

To address this challenge, we propose a novel visual analytics approach (see Fig. 2), in which domain knowledge is used for constructing features capable of effectively distinguishing patterns of interest from other types of behaviour. It is essential to note that these features need to characterise the *behaviour* of relevant variables *on time intervals*, whereas raw data consist of elementary values referring to individual time steps. Hence, feature construction requires knowledge of (a) what aspects of the behaviour are important, e.g., the range of the values or the development trend, and (b) what kinds of computationally derivable aggregate characteristics can represent these aspects. The capability of the features to characterise and differentiate patterns is explored using interactive visualisations, which allow an expert to check whether groups of data items that are similar in terms of the features instantiate the same behaviour patterns and whether groups instantiating different patterns are well separated by the features. The visual aids also allow the expert to select and label groups of representative examples of different pattern types and check the suitability and sufficiency of these examples for generating an automated classifier by means of a machine learning algorithm.

By actively involving a human analyst in the process, our approach achieves flexibility in utilising domain knowledge and accommodating data variations. The interactive visual interface enables simultaneous consideration and labelling of multiple data items, which saves the precious time of the human while allowing creation of a sufficiently large set of data examples for model training.

We evaluated the effectiveness of our approach through case studies focused on detecting trawling activities in fishing vessel trajectories and discriminating different types of offensive play in football (soccer). However, our approach is sufficiently general to be applicable to other domains facing similar challenges.

Related work

The key component of our approach is characterisation of behaviours of time-dependent attributes by expressive features. This kind of task, known as feature

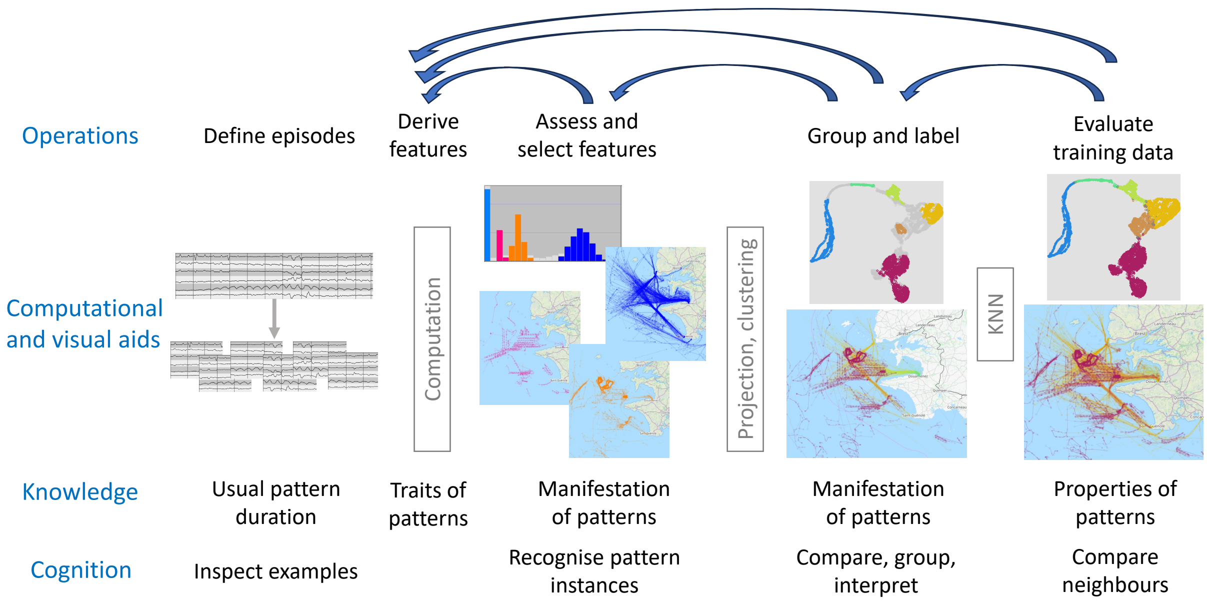

FIGURE 2. A schematic representation of the workflow of the visual analytics approach to involving domain expertise and human cognition in acquisition of training data for creation of a pattern classification model.

engineering [8], is a very important step in the data science pipeline [9]. It is a non-trivial process in which domain knowledge of real-world behaviours to be classified or recognised is translated to appropriate measures summarising and expressing relevant aspects of behaviours of attributes on time intervals. A large number of summary characteristics can be derived from time series [10]. Selecting appropriate ones for specific datasets and tasks is a non-trivial endeavour. Consequently, exploring various feature combinations becomes necessary, facilitated by visual aids to discern how effectively different behaviours are delineated based on chosen features. Despite the abundance of visualization and interaction techniques for temporal data [11], there have been no specific research focusing on visual support for feature engineering.

Another challenge we address in our work is acquisition of labelled data examples for training of machine learning models. This problem is given much attention in the ML research. To reduce the workload of domain experts on providing class labels, the paradigms of active learning [12] and semi-supervised learning [13] have appeared. The former devises strategies and develops algorithms to choose informative unlabeled examples for asking a human "oracle" to provide labels. As discussed in the literature [5], it may be frustrating for the human to repeatedly perform the same kind of routine tasks. Semi-supervised learning designs model training algorithms that leverage a small number of labelled data together with unlabelled data. These algorithms exploit some measure of similarity between data items, assuming that similar items belong to the same class. Both paradigms presuppose that the data already contain features that are characteristic for the classes and sufficient for distinguishing them, which is often not the case. We propose a workflow including feature engineering and subsequent testing of the capability of the chosen candidate features to distinguish the classes. In particular, this allows to verify the key assumption of both active and semi-supervised learning paradigms that similar (in terms of the chosen features) data items belong to the same class. Moreover, in applications where a class may include dissimilar items, our approach allows a domain expert to discover the existence of different variants of data that should be in the same class and refine the categorisation by defining subclasses. Thus, in the first case study, different movement patterns pertinent to trawling activities were discovered.

PROBLEM STATEMENT

Given: A dataset comprising one or more *time series* representing activities or movements of discrete entities or varying states of dynamic processes through values of multiple *numeric attributes*. One time series is a sequence of tuples each including a time stamp and values of the attributes attained at the specified time.

Goal: Prepare training data for generating a machine learning model capable of automated detection of time intervals encapsulating specific types of activities, behaviours, or development trends, termed *patterns*.

Requirements: (1) The time intervals need to be characterised by relevant *features* allowing differentiation of pattern types. (2) A substantial number of labelled pattern examples for subsequent model training need to be acquired.

Challenges: (1) While the values in original data refer to time points, the required relevant features need to characterise behaviours within intervals. (2) Labelled pattern examples either do not exist or are insufficient for model creation. (3) It may not be known in advance what combinations of features can effectively distinguish pattern types.

Background: Domain knowledge is available regarding the characteristics of the patterns of interest, yet there might not be precise mapping of known characteristics to their manifestation in the data. However, it is feasible to represent the data from a time interval in a way that allows a domain expert to recognise the presence or absence of a given pattern or classify the representation as demonstrating a specific pattern.

Example 1 (Maritime traffic monitoring). In the domain of maritime traffic monitoring, there is a task of detecting trawling activities of fishing vessels, which is important for controlling traffic safety and protecting the environment. The available data are trajectories of the vessels having the form of time series where consecutive tuples include *geographic coordinates*, *speed* and *movement direction* of a vessel at different time moments. It is known that trawling typically spans several hours and is characterised by low speed and repeated changes in the movement direction. While specific speed ranges, frequencies of turns, and minimal activity duration are not clearly defined, an expert can recognise trawling by visually inspecting the trajectory shape on a map or in a space-time cube (see Fig. 1).

Example 2 (Playing styles in football). In football (a.k.a. soccer), there is a notion of playing style of a team. The most common concepts occurring in literature and media are direct playing, possession playing, and counter attacking [14]. Direct playing is described as using a small number of passes and prevalence of direct forward passes. Possession playing involves a large number of typically short consecutive passes and slow progression through the midfield. Counter attacking involves the regain of the ball by a defending player close to their goal, followed immediately by a rapid attacking transition towards the opposition's goal. These concepts are quite vague, and domain experts can hardly specify precise boundaries between long and short passes, small and large number of passes, and rapid versus slow development of an attack. However, the shape of the ball trajectory during an attack is indicative of the style of playing in this attack.

It is crucial to note that patterns occur over time intervals. Since data consist of attribute values at time points (instants), detecting pattern manifestations requires transforming the data into higher-level features that characterise activities or developments during intervals. This transformation is a form of temporal abstraction from the initially elementary data.

GENERAL APPROACH

We propose a visual analytics approach to address the problem, enabling collaboration between a domain expert and a computer system. The objective is to derive interval-based features capable of distinguishing behavioural patterns and to generate a sufficient number of labelled pattern examples. This sets the stage for the subsequent development of a machine learning model dedicated to pattern recognition in new data. The approach encompasses the following key components (see Fig. 2):

1) **Representation of time series:** Convert time series into sequences of *episodes*, each of an appropriate duration to encapsulate occurrences of patterns of interest.
2) **Temporal abstraction and feature generation:** Abstract elementary data to interval-based features, capturing pertinent aspects of behaviour or development during the episodes.
3) **Feature assessment and selection:** Explore and assess the utility of the individual features in discriminating patterns. Choose a combination of features that comprehensively represents different aspects of the patterns. If necessary, return to step 2 to generate additional features.
4) **Grouping, examining, and labelling:**
    - Identify groups of similar episodes in terms of the chosen features.
    - Examine whether the episodes within the groups encapsulate the same pattern types. If not, return to the step of feature selection (step 3) or feature generation (step 2).
    - Label the group members as occurrences of known patterns, thereby producing labelled examples.

5) **Evaluation of the training data:** Check the suitability of the produced training data for derivation of a classification model, i.e., whether unlabelled data can be correctly classified based on their similarity to the labelled examples.

The efficacy of the entire process hinges on the apt representation of pertinent dynamics in attribute values through interval-based features. Key aspects, including *value level*, *variability*, *general trend*, *curvature*, and *fluctuations*, can be effectively captured through various computationally derivable interval-based attributes. These attributes encompass:

- **Value level:** minimum, maximum, mean of the attribute values, quantiles, histograms of relative value frequencies (proportions within predefined intervals).
- **Variability:** summary statistics of the positive and negative changes between consecutive time steps, such as the mean of the changes, their variance, amplitude, etc.
- **General trend:** parameters A and B of the trend line $y = Ax + B$; alternatively, the trend can be described by the angle of the trend line inclination.
- **Curvature:** ratio of the sum of consecutive value changes to the difference between the maximum and minimum, sums of positive and negative deviations from the trend line.
- **Fluctuations:** numbers of positive and negative deviations from the trend line, number of intersections of the trend line.

Other potentially useful features and alternative representations for a given aspect may exist [10]. The effectiveness of different features in distinguishing patterns can be assessed by examining the distributions of feature values and visually exploring groups of episodes with low, high, and medium values.

If an initial set of pattern examples is available, it serves as a valuable resource for assessing the consistency and distinctiveness of feature values. Thus, in our case study 1, which is described in the next section, we had results of automated pattern detection by a knowledge-based computer system, which used predefined pattern specifications. Although the system missed numerous pattern occurrences due to data fluctuations, as discussed in the introduction, the successfully recognised pattern instances were used as supporting material for evaluating the effectiveness of candidate interval-based features. Notably, the general approach is versatile and does not hinge on the availability of such supporting material. In the second case study, the methodology was equally applicable, although no prior examples of successful recognition were available.

Regardless of the availability of initial pattern examples, successful feature engineering and selection requires involvement of domain knowledge, including descriptions of patterns of interest and criteria for distinguishing these patterns from the rest of the data. These criteria are translated into appropriate interval-based features, as introduced earlier. In the maritime activities case study, knowledge was encapsulated in formal rules established through prior communication with domain experts. For the football application, essential information was derived from specialised literature, such as [14].

Given the significant reliance of the approach on the cognitive capabilities of a human analyst, interactive visualisations play a crucial role. They empower the analyst to assess the distinctiveness of features, choose representative examples of pattern classes, and evaluate the outcomes of example-based pattern recognition.

We would like to clarify that we aim to introduce a general approach rather than a specific software system. Consequently, we abstain from detailing the user interfaces and interaction tools, focusing instead on presenting a flexible framework that can be implemented in diverse ways to suit different applications and user needs. Rather than testing the use of tools, we assess the effectiveness of our approach by evaluating its ability to produce the expected result: a comprehensive and representative set of labelled examples that capture significant patterns or behaviours in the data. The suitability of this example set for model development is evaluated using the kNN (k-Nearest Neighbour) algorithm [15], which tells us whether new pieces of data encapsulate the same patterns as the labelled examples they are similar to.

CASE STUDY 1: PATTERNS IN MOVEMENT OF FISHING VESSELS

The task in this case study is to detect movement patterns indicating trawling activities in trajectories of 71 fishing vessels operating in the waters northwest of France between October 1, 2015, and March 31, 2016. This is a subset of an openly accessible dataset available at the URL https://zenodo.org/records/1167595 and described in [16].

Data representation. The data are trajectories consisting of all vessel positions recorded during the 6-months period. Obviously, not all parts of the trajectories correspond to the trawling activities. Such long-term trajectories or time series need to be partitioned into segments (episodes) by dividing the time into intervals of appropriate duration based on the expected length of the behaviours or activities under investigation. To achieve this, a sliding time window technique is employed [8]. This method ensures that the data segments overlap partially, preventing the oversight of patterns of interest caused by fragmentation into disjoint parts. In this case study, we segment the trajectories into episodes of 3-hour duration, determined through interactive examination of visually identified trawling movements in sampled trajectories. Employing a sliding window shifted by 1 hour ensures that each occurrence of trawling activity is captured in at least one episode being either fully contained in the time span of the activity or significantly overlapping with it.

Feature engineering. From the expert knowledge encapsulated in the pattern specification, we learn that relevant criteria for recognising trawling are low speed and repeated turns. These criteria need to be represented by appropriate interval-based features derived from the attribute values associated with the vessel positions. The original data include point-based values of speed and movement direction (heading), but both attributes are not ideally suited for deriving expressive interval-based features.

To mitigate potential noise and outliers in point-based speed values, we compute a smoother and more robust measure — average speed over a 5-minute time buffer around each point. It is calculated by dividing the traveled distance by the buffer duration (slightly variable due to irregular time intervals between position recordings). The selection of the buffer duration was made empirically, after experimenting with smaller lengths that failed to eliminate unrealistically high speed values.

The relevant aspect of speed dynamics is the *value level*. We represent it by the combination of the minimum and the third quartile of the point-based values of the smoothed (averaged) speed. The frequency histogram of the third quartiles of the mean speed is shown on the top left of Fig. 3.

The movement direction poses a different challenge due to the cyclic arrangement of the value domain of this attribute, with a cycle length of 360 degrees. While arithmetic differences between low and high values (e.g., between 0° and 359°) are large, the real directional changes are minimal. Deriving suitable interval-based features directly from such attribute proves difficult. Instead, we compute a proxy attribute — the distance of each point from the starting point of an episode. The linear increase of this distance during an episode signifies straight movement, while the presence of turns is reflected in the curvature of the value progression (or, more precisely, the line representing this progression in a line graph). Consequently, we compute a feature expressing *curvature* as the sum of the absolute changes from one point to the next, divided by the difference between the maximal and minimal values. A value of 1 indicates straight movement, while the presence of turns results in higher values for this feature.

Feature assessment and selection. Visualising the frequency distribution of the computed curvature values reveals an extreme left-skewed and long-tailed distribution, making it unsuitable for effective pattern recognition and classification due to low discriminability. The very high values (up to 177) correspond to episodes where vessels remain anchored. In these instances, the accuracy of determining vessel positions diminishes significantly, leading to chaotic scattering of recorded positions around the actual position and creation of a false zigzag-shaped trajectory, as depicted in Fig. 3 (top right). To mitigate the skewness of the feature, a logarithmic transformation is applied to the values. The resulting values now range from 0 (indicating straight movement) to 2.25 (indicating a vessel at anchor), providing a more balanced representation suitable for subsequent analysis.

To assess the adequacy of the speed- and curvature-based features, we employ interactive filtering to select the subset of episodes with low values of speed and values of the curvature logarithm not less than 0.01. A visual examination on the map (Fig. 3, bottom) reveals that the selected subset includes episodes with shapes indicative of trawling movements. However, the dataset also contains numerous instances of vessels entering the port of Douarnenez, displaying low speed and curved trajectories, but unlikely to involve trawling activities. Evidently, the chosen features are insufficient for effectively distinguishing trawling from port entry movements. To address this limitation, we compute additional time series representing the distance to the nearest port, utilising the coordinates of nine ports within the study area. Summary features computed for these time series include the value at the start, value at the end, minimum, and maximum. Employing interactive filtering, we explore the discriminatory potential of these features. The minimum distance to a port emerges as a promising discriminator for episodes of port entry or exit against other episodes. However, due to the highly left-skewed distribution of the feature values, we apply a logarithmic transformation, consistent with our approach for the curvature feature.

Grouping. We employ dimensionality reduction to

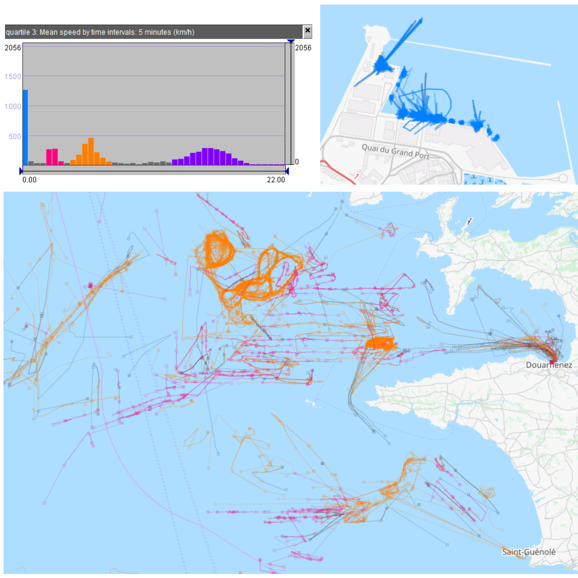

FIGURE 3. Top left: Frequency distribution of the feature '3rd quartile of mean speed in 5 minutes'. Four distinctly coloured groups of bars correspond to intervals in the value range containing the majority of the data. Bottom: Selection of episodes with low speed and high curvature, identified through interactive filtering. Trajectories are colour-coded to match the corresponding histogram bars on the top left.

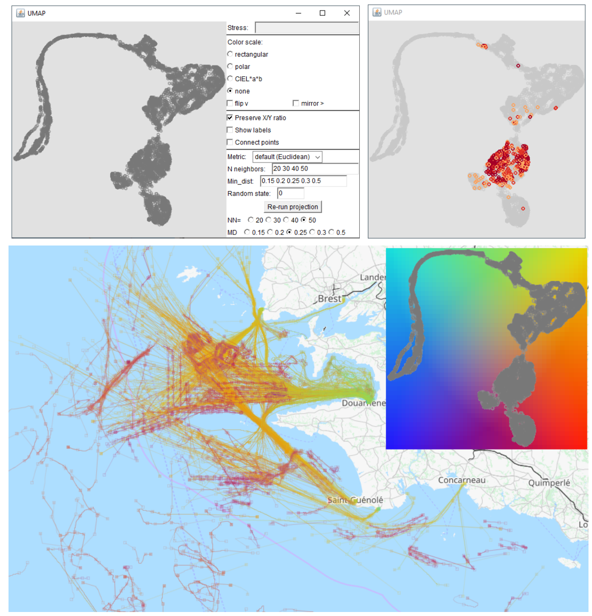

FIGURE 4. Top left: 2D embedding of the episodes based on four features reflecting speed, curvature, and minimal distance to a port. Top right: Episodes with at least 50% of the points recognised as involved in trawling according to the pattern specification are marked in the embedding by colours from orange (50 to 70% of trawling points) to dark red (90% or more), while the remaining episodes are muted (shown in light gray). Bottom: A continuous colour map spread over the projection space is used to colour episodes according to their positions in the projection.

construct a 2D spatial embedding (projection) based on four selected features: the minimum and third quartile of the speed, the logarithm of the curvature, and the logarithm of the minimal distance to the nearest port (Fig. 4, top left). The purpose of the embedding is to expose groups of episodes with similar feature values and enable interactive selection of groups for inspection and eventual labelling. We employ the UMAP method [17] oriented to preserving local neighbourhoods. This means that the method prioritises placing close neighbours in proximity in the embedding, albeit at the expense of potentially distorting distances between non-neighbouring objects. Hence, groups of similar episodes manifest as compact clusters of points in the projection, enabling easy visual detection and interactive selection for inspection. UMAP has two main parameters, number of neighbours to consider *n_neighbors* and the minimal distance between points in the final projection *min_dist*, which controls how tightly UMAP is allowed to pack points together. In our studies, we found the results of UMAP to be highly consistent across various combinations of parameter values. The specific projection demonstrated in Fig. 4 and the following figures has been achieved with *n_neighbors* = 50 and *min_dist* = 0.25.

Examination of groups. To evaluate the effectiveness of the feature combination, we interactively select groups of points from different regions of the projection plot using brushing and observe the shapes of the selected episodes on the map, along with the distributions of their speed characteristics in histograms. Different regions of the projection correspond to distinct movement patterns. This is illustrated in Fig. 4, bottom. A continuous 2D colour scale is spread over the projection space, and the colours corresponding to point positions are used in the map for paining the episodes. Here, we use a 2D colour space named Cube Diagonal Cut B-C-Y-R, which was highly rated in a task-based evaluation study [18]. Despite high overlapping of the lines, the transparency enables seeing that the points in the lower part of the projection plot (purple to red region) represent episodes with shapes characteristic for trawling activities.

Having the results of automated pattern recognition based on a formal specification as supporting material, we conduct an additional test to assess the effectiveness of our selected features. To ensure comparability between the system's results (which pertain to individual points) and our features (which refer to time intervals), we calculate the proportions of recognised trawling points within the episodes. We focus on episodes where at least 50% of the points are recognised as trawling. These episodes are visualised as points coloured from orange to dark red in the top-right projection plot of Fig. 4 while the remaining points are filtered out and shown in light gray. The coloured points form a relatively compact cluster, with a few scattered points distant from the main cluster. Interactive selection and examination of these episodes on the map reveal them to be false positives. This comparison provides additional evidence that the extracted features effectively capture the manifestation of trawling activities in trajectory data and can be used for separating trawling from other types of movements.

Selection and labelling of pattern examples. One possible way to create a sufficiently large set of labelled examples is to select groups of points directly in the projection plot by means of brushing, view the shapes and speed characteristics of the respective episodes, and assign class labels, if the episodes are deemed suitable to serve as class examples. This process needs to be supported by a visualisation that enables seeing the whole group of selected episodes rather than considering them one by one. In this way, a domain expert can create many examples simultaneously. It is important to note that it is, generally, insufficient to create only examples of one or a few patterns of interest, for example, only examples of trawling in this case. There may be multiple ways in which episodes that do not contain the patterns of interest differ from episodes that do. The diversity of the existing patterns can be judged from the distribution of the points in the projection plot. Therefore, it is essential to select and label groups of negative examples (i.e., those with patterns other than the ones of interest) from different regions of the projection where clusters of points exist.

Another effective method for creating class examples is through clustering based on the selected features. Having obtained clusters of episodes with similar combinations of feature values, one can then focus on the core elements within these clusters, interpreting and labelling them as representative examples of different patterns. For instance, in partition-based clustering algorithms like k-means, which we used in our case studies, elements near the cluster centres can be deemed core elements. Alternatively, the representativeness of cluster members can be assessed

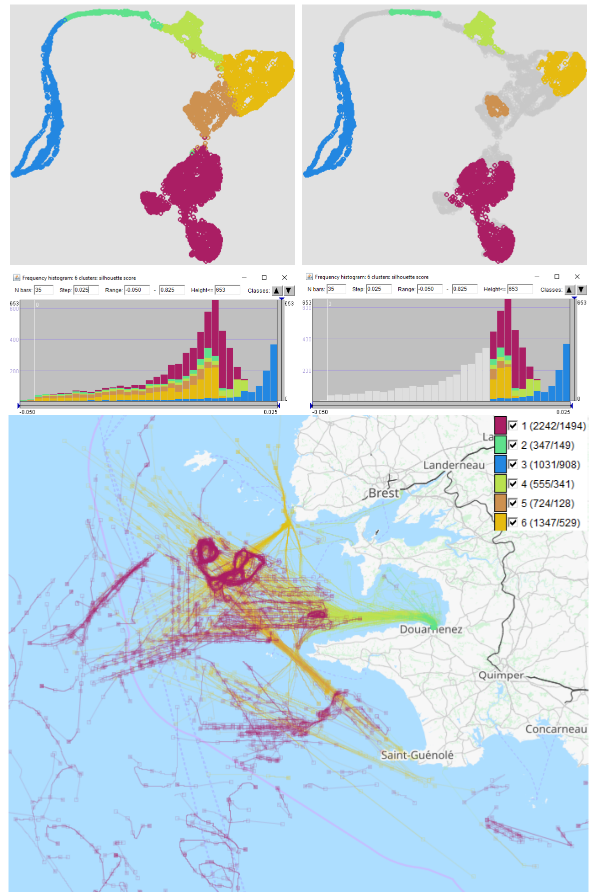

FIGURE 5. Utilising clustering for creation of pattern examples. Top left: The points in the projection plot are coloured according to their cluster membership. Middle left: The histogram of the silhouette scores of the cluster members. Top right: The points in the projection with the silhouette scores below 0.55 are filtered out (shown in light grey). Middle right: The appearance of the histogram after the filtering. Bottom: The episodes belonging to the cluster cores, i.e., having silhouette scores of at least 0.55, are painted in the colours of the clusters. The numbers in the legend are the total cluster sizes followed by the counts of the members satisfying the filter.

using silhouette scores, which indicate how similar an object is to other members within its cluster compared to members in other clusters. A high score suggests that the object can be considered representative for its own cluster. Determining the appropriate number of clusters, which is a parameter in partition-based clustering algorithms, can be done by assessing the point distribution in the projection and fine-tuning through small adjustments of the parameter value.

Our experience indicates that the clustering approach is more effective than manually selecting representative examples of patterns using the projection plot. Besides, since any low-dimensional projection of higher-dimensional data introduces distortions, neighbouring points in the projection may not be as similar as their proximity suggests. Consequently, when a group of points is selected through brushing, there is no complete assurance that all corresponding episodes are very similar to each other. Such interactively selected episode groups must be carefully inspected, increasing the workload for the expert.

The use of clustering for selection of representative pattern examples is illustrated in Fig. 5. Cluster 1 (purple) includes episodes with shapes and speeds characteristic for trawling activities. As cluster cores, we select the subsets of cluster members whose silhouette scores are not less than 0.55. However, upon filtering episodes based on the silhouette scores, we observe that the cores of cluster 2 (light blueish-green) and 5 (light brown) consist of considerably fewer members than the cores of the other clusters. Additionally, a compact group of points from cluster 1, positioned to the left of the large purple cluster in the projection plot, has been almost entirely filtered out (Fig. 5, top right). Interactively selecting this group of points reveals (Fig.6) that they correspond to notable instances of looping movements, which should be included as examples of trawling patterns. So, we use interactive selection and deselection operations in the projection plot and silhouette score histogram (to exclude instances with too low scores) to enhance the subsets of chosen cluster representatives. The ultimate outcome of selecting and labelling representative examples of different patterns is displayed in Fig.7.

Evaluation of the training dataset. As a tool for evaluation, we employ the kNN algorithm [15]. Being applied to unlabelled data, it shows whether their similarity to the examples in terms of the features implies the presence of the same patterns as in the examples.

As we have created quite a large number of labelled examples, we take $k = 50$, where k is the number of nearest neighbours used for classification of each data item. The results are demonstrated in Fig. 8. The total number of identified episodes of trawling is 2261, of which 1600 have been previously selected as pattern examples and the remaining 661 have been recognised as trawling due to their similarity to the examples regarding the constructed four distinctive features. The trajectories of these 661 episodes visible in Fig. 8 (top) closely resemble the chosen pattern examples (Fig. 7, left), taking the anticipated shapes of vessel trajectories during trawling. Although the misclassification of port entering episodes as trawling could not be entirely avoided, there are only 19 such episodes among the

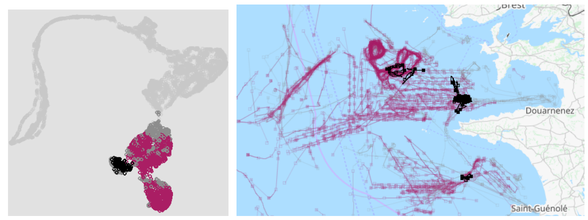

FIGURE 6. Creation of additional pattern examples through interactive selection of points in the projection.

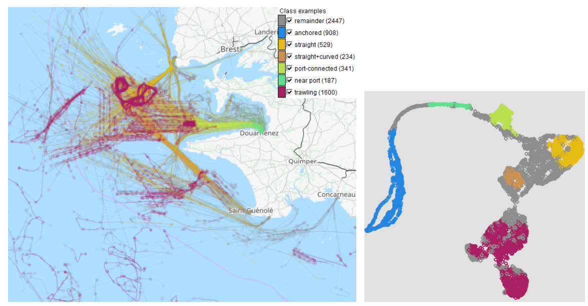

FIGURE 7. Class examples supplied with semantically meaningful labels.

KNN results. In 76 episodes, the movement patterns were not recognised with sufficient certainty. These points are depicted in dark grey in the projection plot in Fig.8 (bottom). Their location at cluster boundaries indicates that they combine features of two pattern types. The overall success of the kNN results indicates that the features and examples are suitable for training a more sophisticated and more accurate classification model by means of state-of-the-art ML algorithms.

Concluding notes. The outcomes of our workflow can not only contribute to the development of a classifier but also provide valuable insights for enhancing existing specifications employed in automated knowledge-based pattern recognition [19]. While the current specifications identified only 530 episodes with at least 50% of trawling points, and 12 of these were port entering episodes, our approach significantly expanded this capability, correctly recognising 2242 trawling episodes. Remarkably, our method unveiled three distinct subtypes of trajectory shapes corresponding to trawling: wide curves, tight loops, and slow straight movements alternating with 180° turns. The latter subtype remained undetected by the specification-based system due to the predefined upper limit on the duration of straight movements. Thus, our visual analytics approach not only facilitated the detection of expected patterns but also enabled the discovery of unexpected ones [20].

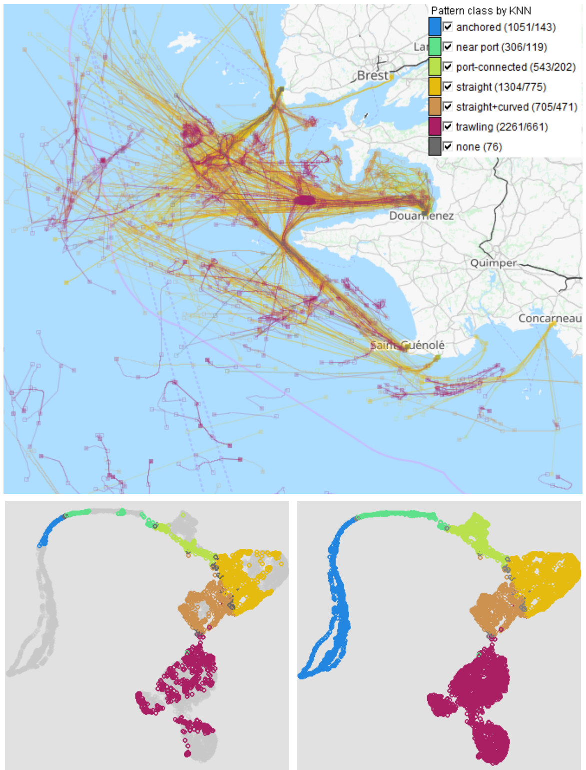

FIGURE 8. Classification of the movement patterns using KNN algorithm with $k = 50$. Top: The episodes classified based on their feature similarity to the pattern examples are shown on the map. Bottom left: Points in the projection representing the classified episodes are colour-coded according to their class membership. Pattern examples created earlier are subdued (coloured in light grey). Dark grey points represent episodes where movement patterns were not recognized with sufficient certainty (at least 60%). Bottom right: Coloured points signify both pattern examples and patterns identified by the KNN algorithm.

CASE STUDY 2: STYLES OF ATTACKS IN FOOTBALL

Data representation. In this case study, our focus is on episodes from two football (soccer) matches, corresponding to alternating ball possessions by each team. The objective is to discern various playing styles, such as direct versus possession-preserving strategies. To ensure meaningful analysis, we filtered out episodes with duration below 5 seconds, resulting in a dataset of 380 episodes.

While each episode is characterised by a multitude of time-variant attributes, encompassing the positions and velocities of all players and the ball, the shape of the ball trajectory within an episode distinctly reflects the manner in which the attack unfolds. The ball trajectory is easy to visualise and to interpret, which is essential for the successful application of our approach. Consequently, we will employ features that capture the essence of ball movements for the differentiation and classification of playing styles.

To facilitate consistent comparison among episodes irrespective of the orientation of the teams' goals on the pitch (left or right), we standardised the coordinates in the ball trajectories (in the context of this case study, the term "goal" refers to the goal gates and underlying areas on two opposite sides of the pitch). The transformation aligns the direction of the attack with the Y-axis. The pitch is depicted in a vertical orientation, with the goal of the team in possession of the ball at the bottom and the target goal, i.e., the goal being attacked, at the top. This transformation is illustrated in Fig. 9.

Feature engineering. Upon careful consideration of various instances of episodes and consulting the literature (such as [14]), we observe that distinct styles of attacks are discernible based on features derived from the time series of the X- and Y-coordinates of the ball. Notably, wide amplitude and significant lateral ball movements (along the X-axis in the transformed coordinate system) suggest a team's effort to retain possession through repeated passes among defenders and/or midfielders. Conversely, characteristics like low variation in the X-coordinate and an increase in the Y-coordinate from the beginning to the end are indicative of direct playing. Counter-attacks are identifiable by predominantly forward (upward in the transformed coordinate system) ball movements, starting in proximity to the goal of the team in possession. Consequently, we derive the following 8 features for the episodes:

- Range of the X-coordinate, calculated as the difference between the maximum and minimum.
- Logarithm of the curvature of the X-coordinate variation. The logarithmic transformation compensates for the skewness in the distribution of curvature values.
- Initial value of the Y-coordinate at the episode's beginning.
- Change in Y from the beginning to the end of the episode, representing the advance of the ball up the pitch.
- Angle of the trend line of the Y-coordinate variation, reflecting the overall direction and speed of the ball movement (forward or backward).
- Coefficient of linear correlation between the Y-coordinate and time, indicating variations in the directions of ball movements.

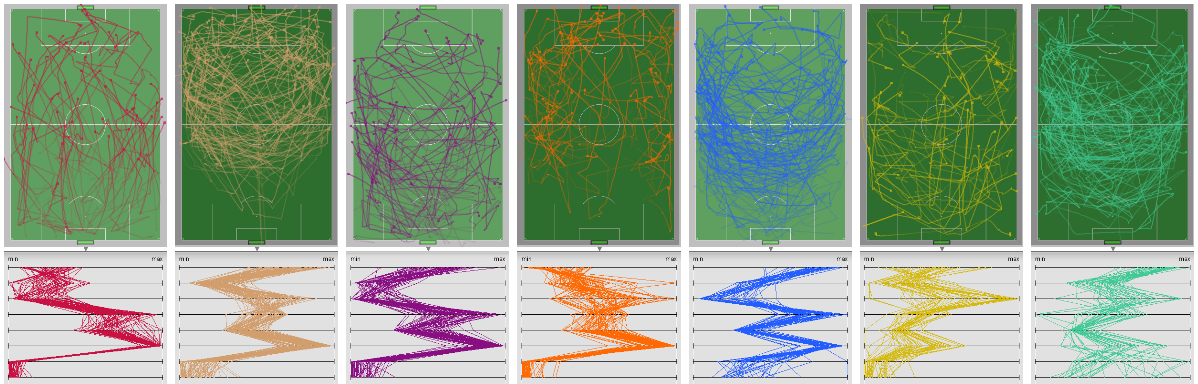

FIGURE 9. Clusters of football episodes. The trajectories are shown on the pitch maps. The profiles of the cluster members in terms of the features are shown in the parallel coordinates plots below the maps. Each axis represents one feature.

- Sums of (a) positive and (b) negative deviations of Y values from the trend line, reflecting forth and back ball movements.

Grouping and examination. As in the first case study, clustering of episodes by similarity of their feature values serves as a tool for generating representative examples of various playing styles. We again employ UMAP to obtain a 2D embedding based on the same features to aid in determining an optimal number of clusters and systematically evaluate the impact of altering this number. For this purpose, we paint dots in the embedding in the colours of the clusters containing the corresponding episodes, as illustrated in Fig.9. After careful consideration, we decide to use the outcome with 7 clusters. The positions of the cluster members within the 2D projection are depicted in Fig.10, left. The projection was produced by UMAP with *n_neighbors* = 16 and *min_dist* = 0.25. The parameters of the embedding can be seen in the middle of Fig.10. In contrast to the first case study, the projection plot does not manifest compact, densely populated clusters that are well separated. The football case is different due to a considerably smaller number of episodes and a lack of partial overlap between them, resulting in each football episode having few, if any, close neighbours.

Selection and labelling of pattern examples. We calculate the silhouette scores of the cluster members. The diversity among the episodes, even within clusters, is evident in the relatively low values of the silhouette scores of the cluster members, ranging from -0.13 to 0.497. We take the cluster cores, encompassing the members with scores of at least 0.24 (Fig.10, right), as a base for constructing groups of representative examples for distinct playing styles. Evaluating the core members of each cluster through visualisations, we interpret the patterns and assign meaningful labels.

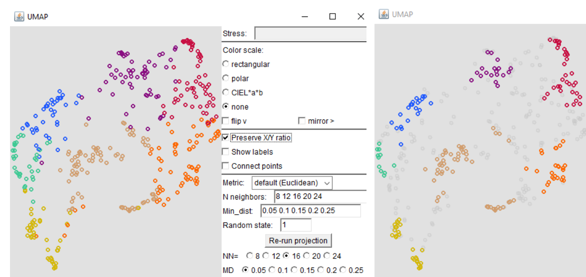

FIGURE 10. Left: Projection with dots coloured according to the cluster membership. Middle: Parameters of the embedding and the projection display. Right: The dots representing episodes with low cluster silhouette scores are muted (coloured in light gray).

The examples of playing styles defined in this manner are shown in Fig.11. The groups of examples received the following labels: direct play (red), keeping possession in attack (beige), slow advance (purple), wing attacks (orange), wing attacks after buildup (blue), backward passes (yellow), and keeping possession in centre (turquoise). It can be noted that the data allow more refined categorisation of playing styles compared to the few basic styles appearing in the literature and media. Particularly, possession playing appears in several variants differing in the pitch area where the ball mostly moves: close to the target, in the centre, or in the buildup area close to the own goal. There are also short episodes, when teams did not manage to maintain ball possession long enough to demonstrate one of the basic playing styles. Thus, the group labelled “wing attacks” consists of such short episodes.

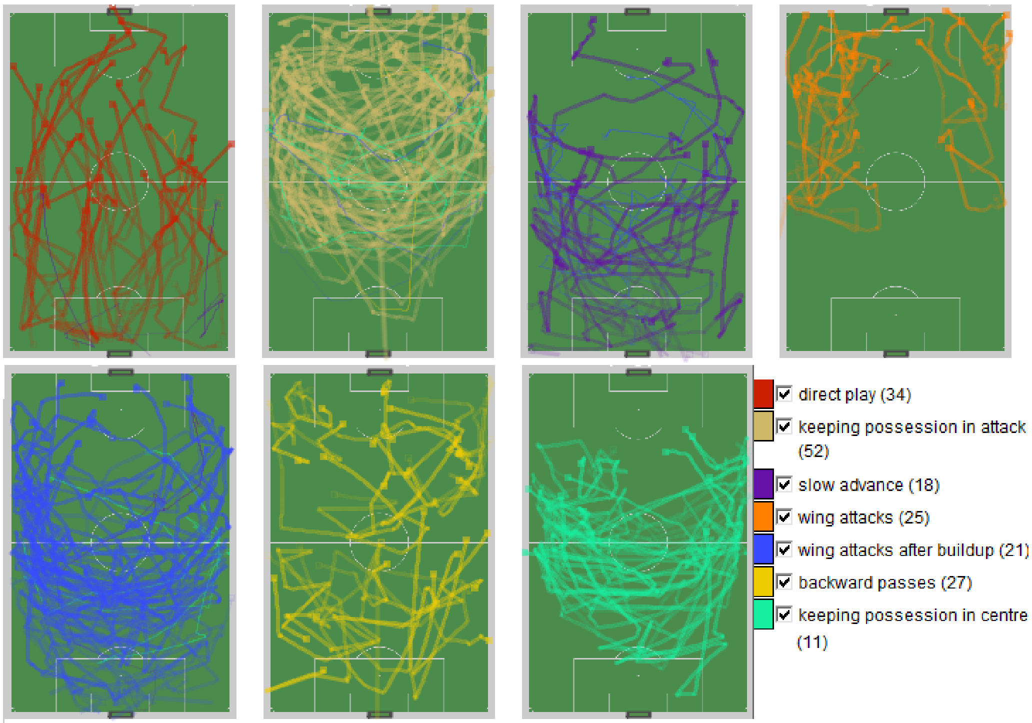

FIGURE 11. Examples of different playing styles in football episodes.

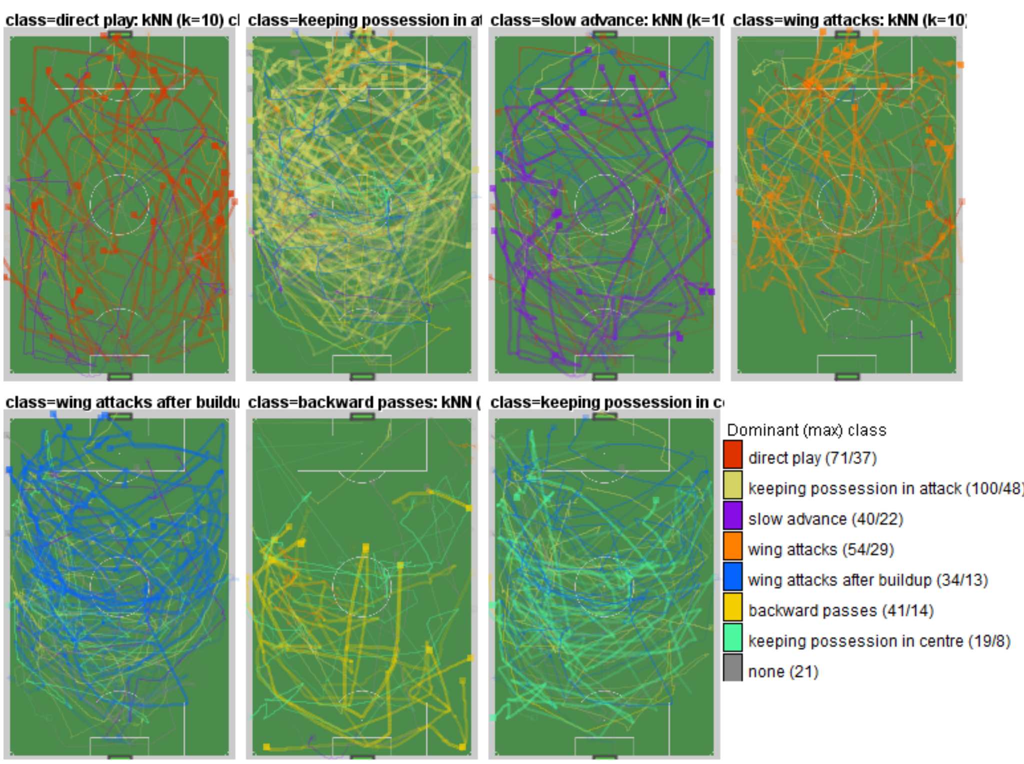

FIGURE 12. Instances of playing patterns recognised by means of KNN with k = 10 using the provided pattern examples.

Evaluation of the training set. Like previously, we utilise KNN classification (k = 10) to assess the suitability of the examples and selected features for developing a ML model. We take a smaller value for k than in the first case study because we have here a smaller dataset and, accordingly, smaller groups of similar episodes. The maps in Fig. 12 portray the episodes classified as instances of the specified playing styles, in addition to the provided examples. Each map contains episodes with non-zero weights assigned to the respective classes by the KNN algorithm. The trajectories of the ball in the episodes are colour-coded based on the class with the highest weight. It can be noticed that the maps in Fig. 12 look messier than in Fig. 11. This is because one map may include differently coloured trajectories, as it shows not only the members of one class according to the kNN classification but also other trajectories with non-zero weight of this class albeit being members of other classes. However, trajectories with lower weights of the represented class are depicted with reduced opacity. The legend in the lower right corner indicates the overall sizes of the classes, including the examples, followed by the number of episodes additionally recognised by the KNN algorithm. Twenty-one episodes remained unclassified, as none of the class weights surpassed 50%, signifying that the pattern could not be identified with sufficient certainty.

Discussion. While the first case study aimed at detecting manifestations of concrete activities in data, the playing styles we aimed to recognise in the second case study were rather vague abstract concepts, and it was unclear how they could manifest in the data. We found that particular properties of the ball movements are indicative of the manner in which an attack was developing, and we used them to reveal several meaningful patterns. It turned out that the number of distinct patterns exceeds the number of the commonly recognised playing styles, but most patterns can be seen as variations of these playing styles. Thus, ‘keeping possession in attack’, ‘wing attacks after buildup’, and ‘keeping possession in centre’ are variations of the possession-keeping style. The use of backward passes to the goalkeeper can also be seen as a tactic to maintain possession, although backward passes are not explicitly mentioned in the descriptions of the possession-keeping style occurring in media. The pattern ‘wing attacks’ is a variant of the direct playing style, whereas the pattern labelled ‘direct play’ includes also counterattacks.

Concluding notes. In this case study, domain knowledge allowed us to determine relevant properties of ball movements that can pertain to different playing styles. Using visual analytics, we were able to detect indications of these playing styles in data, but we also observed high diversity of the ways to develop attacks and discovered several variations of realising the concepts of the playing styles.

DISCUSSION AND CONCLUSION

Our approach stands out for its primary objective: rather than merely analysing a specific dataset, it aims at producing a comprehensive set of labelled representative examples of data encapsulating various behavioural patterns. Ultimately, these examples are intended to serve as the foundation for training a classification model capable of identifying such patterns in new datasets.

The approach proves particularly valuable in scenarios where patterns of interest lack precise definitions, which is a common challenge in many real-world applications. Furthermore, it addresses the complexity of unknown possible manifestations of these patterns in data. For instance, football experts, accustomed to differentiating playing styles through direct observation of a game, do not know how these styles translate into trajectory data. Similarly, maritime traffic managers know how fishing vessels typically move while trawling but have quite little understanding of the corresponding properties of the vessel tracks.

A crucial step involves temporal abstraction [11, Section 6.3], where elementary attribute values referring to individual time steps are transformed into interval-based summary features expressing the way in which the elementary values vary. This adaptation is essential since behaviours inherently occur within time intervals, whereas the raw data pertains to discrete time points. We propose a catalogue of summary attributes meant for expressing different aspects of distribution and variation of numeric values within intervals. An expert or data analyst can choose potentially suitable attributes corresponding to the relevant facets by which distinct behaviours may differ from each another.

It is worth noting that, although the case studies demonstrate the approach in application to movement data, the features employed are not exclusive to movement data but can be derived from any numeric time series. This means that the approach can find general applicability across diverse domains, describing various behaviors, activities, or dynamic phenomena through multivariate time series data. For example, we successfully applied it to time series of population health and mobility data during the COVID pandemic. The only prerequisite for using the approach is the ability to generate expressive visualisations of groups of similar episodes. Thus, working with the COVID morbidity and mobility data, we visualised the temporal progression of the attribute values in groups of episodes using time series graphs.

Designed to enable involvement of domain experts, the approach aids in structuring and refining their prior knowledge and empowers them to explore how expected patterns manifest in the data by assessing different characteristics of variation within time intervals. The expert knowledge and evolving understanding of the data semantics are captured in the form of data-derived relevant features and labelled examples, which become vital inputs for training a machine learning model. This integration of human expertise and machine learning not only enhances model interpretability but also ensures alignment with human domain knowledge.

In summary, the main properties of our approach are its focus on preparation of training data for model building, adaptability in dealing with loosely defined patterns, incorporation of temporal abstraction, facilitation of expert involvement, and the extraction of domain knowledge in the form of features and labeled examples of behavioural patterns. The key role in the approach belongs to interactive visualisations enabling human analysts to apply their cognitive capabilities.

On the flip side, our approach does have its weaknesses, largely stemming from its reliance on human involvement. Firstly, while considering groups rather than individual data items reduces the time burden on human experts, significant time investment is still required. Enhancing computational support for data grouping could help alleviate this issue. Secondly, human error and the potential oversight of crucial details are inherent risks. To address this, the approach needs to be implemented so as to facilitate, encourage, and perhaps even enforce visual and computational evaluations of work outcomes. Thirdly, the design of suitable visualisations and interaction tools, along with the seamless integration of computations into interactive visual interfaces, is crucial for the successful application of the approach. This places a significant responsibility on software designers and engineers.

In our paper, we deliberately avoided delving into implementation specifics, focusing on presenting the approach itself. To test it and provide illustrations for the paper, we utilised an in-house multi-functional visual analytics system. While this system is robust and fully supports the workflow, we acknowledge that its complexity may pose a challenge for domain experts working alone. However, high complexity is a common trait among software systems designed for handling complex tasks. Ultimately, the most effective use of such systems is achieved when domain experts collaborate with data scientists.

In the future, the results of our work can be utilised in developing an intelligent system assisting domain specialists by suggesting appropriate features capable of reflecting various facets of behaviours. Furthermore, given a set of selected features, the system will be able to extract and characterise different behavioural patterns, streamlining the process of pattern recognition and example generation.

ACKNOWLEDGMENTS

This work was supported by Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the *Lamarr Institute for Machine Learning and Artificial Intelligence* (Lamarr22B), and by EU in project *CrexData* (grant agreement no. 101092749).

REFERENCES

1. Benkert, M., Gudmundsson, J., Hübner, F. & Wolle, T. Reporting flock patterns. *Computational Geometry*. **41**, 111-125 (2008)
2. Pitsikalis, M. & Artikis, A. Composite Maritime Event Recognition. *Guide To Maritime Informatics*. pp. 233-260 (Springer, 2021)
3. Artikis, A., Sergot, M. & Paliouras, G. An Event Calculus for Event Recognition. *IEEE Transactions On Knowledge And Data Engineering*. **27**, 895-908 (2015)
4. Mantenoglou, P., Artikis, A. & Paliouras, G. Online Probabilistic Interval-Based Event Calculus. *ECAI 2020 - 24th European Conference On Artificial Intelligence*. **325** pp. 2624-2631 (2020)
5. Andrienko, N., Andrienko, G., Adilova, L. & Wrobel, S. Visual Analytics for Human-Centered Machine Learning. *IEEE Computer Graphics And Applications*. **42**, 123-133 (2022)
6. Andrienko, N., Andrienko, G., Fuchs, G., Slingsby, A., Turkay, C. & Wrobel, S. Visual analytics for data scientists. (Springer, 2020)
7. Andrienko, N., Andrienko, G., Miksch, S., Schumann, H. & Wrobel, S. A theoretical model for pattern discovery in visual analytics. *Visual Informatics*. **5**, 23-42 (2021)
8. Duboue, P. *The Art of Feature Engineering: Essentials for Machine Learning*. Cambridge University Press; (2020).
9. O'Neil, C. and Schutt, R. *Doing data science: Straight talk from the frontline*. O'Reilly Media, Inc.; (2013).

10. Lubba, C., Sethi, S., Knaute, P., Schultz, S., Fulcher, B. & Jones, N. Catch22: CAnonical Time-Series CHaracteristics: Selected through Highly Comparative Time-Series Analysis. *Data Mining and Knowledge Discovery*. **33**, 1821-1852 (2019)
11. Aigner, W., Miksch, S., Schumann, H., Tominski, C.*Visualization of Time-Oriented Data, 2nd edition*. Springer, London (2023).
12. Monarch, R., Munro, R., Manning, C.D.*Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI*. Manning (2021).
13. van Engelen, J.E., Hoos, H.H. A survey on semi-supervised learning. *Machine Learning*. **109**, 373–440 (2020).
14. Plakias, S.; Moustakidis, S.; Kokkotis, C.; Tsatalas, T.; Papalexi, M.; Plakias, D.; Giakas, G.; Tsaopoulos, D. Identifying Soccer Teams' Styles of Play: A Scoping and Critical Review. *Journal of Functional Morphology and Kinesiology*. **8**, (2023)
15. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Transactions On Information Theory*. **13**, 21-27 (1967)
16. Andrienko, N. & Andrienko, G. Visual Analytics of Vessel Movement. *Guide To Maritime Informatics*. pp. 149-170 (Springer, 2021)
17. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv: 1802.03426*, (2020)
18. Bernard, J., Steiger, M., Mittelstädt, S., Thum, S., Keim, D. & Kohlhammer, J. A survey and task-based quality assessment of static 2D colormaps. *Visualization And Data Analysis*. **9397** pp. 93970M (2015)
19. Katzouris, N., Paliouras, G. & Artikis, A. Online Learning Probabilistic Event Calculus Theories in Answer Set Programming. *Theory And Practice Of Logic Programming*. **23**, 362-386 (2023)
20. Thomas, J. & Cook, K. Illuminating the Path: The Research and Development Agenda for Visual Analytics. (IEEE Computer Society, 2005)

Dr. Natalia Andrienko is a lead scientist at Fraunhofer Institute for Intelligent Analysis and Information Systems, Area Chair and Co-PI of the Lamarr Institute for Machine Learning and Artificial Intelligence, and professor at City University of London. Results of her research have been published in two monographs, "*Exploratory Analysis of Spatial and Temporal Data: a Systematic Approach*" (2006) and "*Visual Analytics of Movement*" (2013), and in a textbook "*Visual Analytics for Data Scientists*" (2020). Natalia Andrienko is an associate editor of *Visual Informatics*. Contact her at natalia.andrienko@iais.fraunhofer.de.

Dr. Gennady Andrienko (www.geoanalitycs.net) is a lead scientist responsible for visual analytics research at Fraunhofer Institute for Intelligent Analysis and Information Systems, Co-PI of the Lamarr Institute for Machine Learning and Artificial Intelligence, and professor at City University of London. Gennady Andrienko is an associate editor of *Information Visualization* and *International Journal of Cartography*. Contact him at gennady.andrienko@iais.fraunhofer.de.

Dr. Alexander Artikis is an Associate Professor in the University of Piraeus (GR), and a senior Research Associate in NCSR 'Demokritos' (GR), leading the Complex Event Recognition group (https://cer.iit.demokritos.gr). He has been working for more than 20 years in the field of Artificial Intelligence, publishing papers in the top journals and conferences of the field, such as the AI journal, ACM ToCL, ACM TAAS, ACM TIST, IEEE TKDE, VLDBJ, Machine Learning journal, JAIR and KR. Contact him at a.artikis@unipi.gr.

Periklis Mantenoglou is a PhD candidate at the National and Kapodistrian University of Athens (GR), and a research associate at NCSR 'Demokritos' (GR). His research interests include stream reasoning, logic programming, event processing and probabilistic reasoning. Contact him at pmantenoglou@iit.demokritos.gr.

Dr. Salvatore Rinzivillo is a Senior Researcher at the Knowledge Discovery and Data Mining Laboratory, Institute of Science and Technology of Information, National Research Council, serving as Principal Investigator for the Applied Data Science group. SR's research interests are at the intersection of spatio-temporal data analysis, network analysis, and visual analytics. Recently, his research activities have extended to include a focus on developing methodologies and interfaces for explainable artificial intelligence (XAI). Contact him at rinzivillo@isti.cnr.it