CHAPTER SIX

SEASONAL FOURIER FEATURES

6.1 OVERVIEW

In this chapter, the concept of extracting meaningful features from a time series is investigated. The chapter starts by defining the difference between the concept of whole clustering and subsequence clustering. It continues by exploring a fundamental pitfall inherent when using subsequence clustering to analyse time series. This is motivated at the hand of an experiment presented by Keogh [29] and a worked-out visual example. A key feature extraction method, that will extract the Seasonal Fourier Features (SFF) is presented in section 6.4, which will overcome the disadvantage of using subsequence clustering. The chapter concludes by defining how this SFF is used in a post-classification change detection algorithm to detect change in time series.

6.2 TIME SERIES ANALYSIS

A time series is a sequence of measurements, typically recorded at successive time intervals [191]. Time series have a distinct natural temporal ordering. This induces a high correlation between measurements taken at a shorter interval from a system, when compared to measurements taken at a longer interval from the same system. Time series analysis comprises methods for analysing time series to extract statistics and underlying characteristics. Several different types of analysis can be applied to time series and are categorised as: exploration, description, prediction and forecasting.

1. Exploration provides in-depth information on serial dependence and any cyclic behaviour patterns within time series. The time series can also be graphically examined to observe any salient characteristics.

2. Description provides information of underlying structures hidden within the time series. Algorithms were developed to decompose time series into several components to examine any hidden trends, seasonality, slow and fast variations, cyclic irregularities and anomalies.

3. Prediction provides information on any near future event in the time series and can be used as feedback to control a system's behaviour that is providing the data points of the time series.

4. Forecasting uses statistical models to generate variations of the time series to observe alternative possible events that might occur in the future.

Clustering is the most frequently used exploration tool in data mining algorithms. The vast quantities of important information typically hidden in time series have attracted substantial attention [29]. Clustering is used in many algorithms as either: rule discovery [192], indexing [193], classification [194], prediction [195], or anomaly detection [196]. Clustering of time series is broadly divided into two categories: *whole clustering* and *subsequence clustering* [29].

Whole clustering: Whole clustering is similar to the conventional clustering of discrete objects. Each time series is viewed as an individual discrete object and is thus clustered into groups with other time series. □

Subsequence clustering: Subsequence clustering is when multiple individual time series (subsequences) are extracted with a sliding window from a single time series. Let $\mathbf{x}$, $\mathbf{x} = [\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_{\mathcal{I}}]$, denote a time series of length $\mathcal{I}$. A subsequence extracted from time series $\mathbf{x}$ is given as

$$\mathbf{x}_p = \left(\vec{x}_p,\ \vec{x}_{p+1},\ \ldots,\ \vec{x}_{p+Q-1} \right), \tag{6.1}$$

for $1 \leq p \leq \mathcal{I}$-$Q$+1, where Q is the length of the subsequence. The sequential extraction of subsequences in equation (6.1) is achieved by using a temporal sliding window that has a length of Q and position p, $p \in \mathbb{N}_0$, that is incremented with a natural number $\mathbb{N}$ to extract sequential subsequences $\mathbf{x}_p$ from $\mathbf{x}$. This set of subsequences are clustered into groups, similar to how *whole clustering* clusters an entire time series. □

6.3 MEANINGLESS ANALYSIS

Recently the data mining community's attention was drawn to a fundamental limitation in the clustering of subsequences that are extracted with a sliding window from a time series [29]; the sliding window

causes the clustering algorithms to create meaningless results. This is due to the fact that clusters extracted from the subsequences are forced to obey a certain constraint that is pathologically unlikely to be satisfied by any data set. The term meaningless originates from the effect of creating random clusters when applying a clustering algorithm to such subsequences [29].

It should be noted that it is well understood that clustering in a high-dimensional feature space usually produces useless results if proper design considerations are not followed [197, 198]. For example, the K-nearest neighbour algorithm produces fewer useful clusters in higher dimensions. This is because the ratio between the nearest neighbour and the average neighbour distance rapidly converges to one in higher dimensions. However, the analysis on time series usually results in high dimensionality, which typically has a low intrinsic dimensionality [199]. This is not the limitation that will be discussed in this chapter.

Keogh and Lin [29] made a surprising claim, which called into question dozens of published results. The problem identified lies in the way the features are extracted from the sliding window when presented to the clustering algorithm. This claim is supported by the following experiment.

Experiment presented in [29]: The variability in the clusters formed will be tested using the same clustering design considerations and methodology on different data sets containing time series. It is shown that any partitional or hierarchical clustering algorithm would suffice in this experiment, and under this assumption the K-means was used for its robustness in forming reliable clusters. The K-means clustering algorithm forms clusters, which are used to define a set of functions.

Let $\vartheta(a) = \{\vartheta^1(a), \vartheta^2(a), \ldots, \vartheta^K(a)\}$ denote the cluster centroids derived with the K-means algorithm from the first data set.

Let $\vartheta(b) = \{\vartheta^1(b), \vartheta^2(b), \ldots, \vartheta^K(b)\}$ denote the cluster centroids derived with the K-means algorithm from the second data set.

Let $D_{\text{ed}}(\vartheta^i, \vartheta^j)$ denote the Euclidean distance between two cluster centroids. The distance metric $D_{\text{ed}}(\vartheta^i, \vartheta^j)$ determines the shortest possible distance for an one-to-one mapping of two sets of centroids $\vartheta(a)$ and $\vartheta(b)$.

The difference between the two sets of cluster centroids is defined as

$$D_{\mathcal{M}}(\vartheta(a), \vartheta(b)) = \sum_{i=1}^{K} \min_{j} [D_{\text{ed}}(\vartheta^i(a), \vartheta^j(b))]. \tag{6.2}$$

The consistency of a clustering algorithm to form similar sets of clusters is measured if the first data set used to find cluster centroids $\vartheta(a)$ and the second data set used to find cluster centroids

$\vartheta(b)$ is the same data set. A more important measurement is to determine the similarity between the centroids when they are not the same data set.

Keogh and Lin [29] proposed a clustering meaningfulness index as

$$C_{\mathcal{M}}(\vartheta(a), \vartheta(b)) = \frac{D_{\mathcal{M}}(\vartheta(a), \vartheta(a))}{D_{\mathcal{M}}(\vartheta(a), \vartheta(b))}. \tag{6.3}$$

The clustering meaningfulness index measures the similarity between two data sets' clusters despite the fact that two different data sets are used.

Intuitively, if proper clustering design considerations were applied the numerator in equation (6.3) should converge to zero. In contrast to this statement, if the data sets are unrelated, then the denominator should tend to a large number. This in effect naturally makes the clustering meaningfulness index $C_{\mathcal{M}}(\vartheta(a), \vartheta(b)) \to 0$.

The results produced in this experiment were unexpected. When a random walk data set was compared to a stock market data set, the clustering meaningfulness index averaged between 0.5 and 1 when *subsequence clustering* was applied to the time series. This means that if clustering was performed on the stock market data set, the centroids derived could be re-used for the random walk data set and the difference in clustering results could not be observed.

The same was not true when *whole clustering* was used on these two data sets. The clustering meaningfulness index converged to zero when the stock market data set and random walk data set were clustered using a *whole clustering* approach. Several additional experiments were conducted in [29] to motivate this behaviour as a property of the sliding window. □

The sliding window causes the clustering algorithm to create meaningless results, as it forms sine wave cluster centroids regardless of the data set, which clearly makes it impossible to distinguish one data set's clusters from another. Furthermore, the sine waves within the cluster centroids are always out of phase with each other by exactly $1/K$ period [29]. The inability to produce meaningful cluster centroids revealed a new question: how do the cluster centroids obtain this special structure [29]? In this section a visual example is shown to illustrate why the clustering algorithm produces meaningless results.

Visual example: Assume a triply modulated cosine function, which is given as

$$x_i = \mu_i + \alpha_i \cos(2\pi f i + \theta_i), \tag{6.4}$$

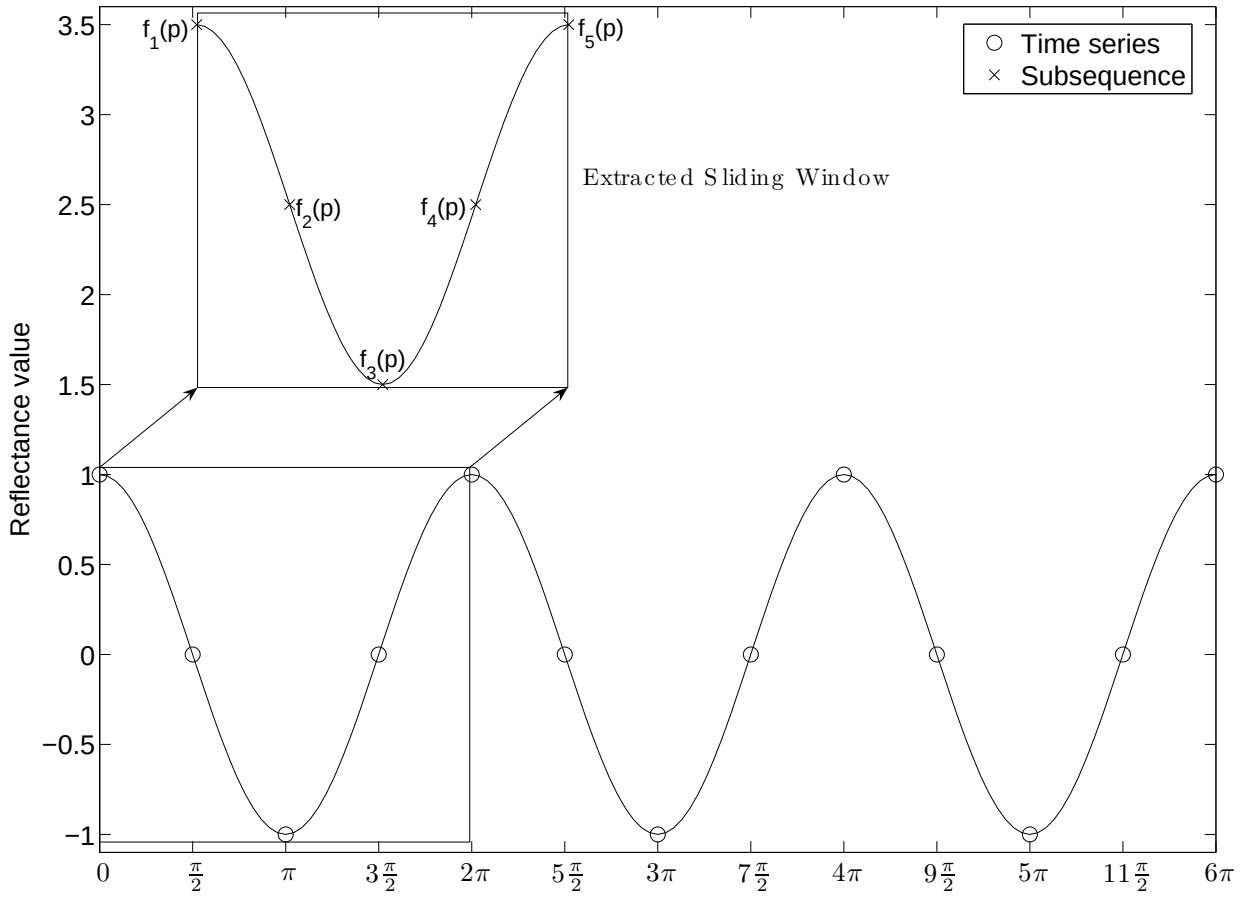

FIGURE 6.1: The five feature points, separated by a period of $\frac{\pi}{2}$, are extracted from the sliding window, and is denoted by the set $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$.

where the mean μ_i, amplitude α_i, frequency f, and phase θ_i are fixed for all time increments in this example. A visual plot of this triply modulated cosine function is shown in figure 6.1. A sliding window is placed on the time series with features extracted from the window at multiples of $\frac{\pi}{2}$ of the period.

The five features are extracted at interval $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$ from the sliding window and are denoted by $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$. The position of the sliding window is denoted by the variable p, $p \in \mathbb{N}_0$. This is mathematically expressed as

$$\begin{aligned} \mathbf{x}_p &= \Big(f_1(p),\ f_2(p),\ f_3(p),\ f_4(p),\ f_5(p)\Big) \\ &= \Big(x_{p\pi/2},\ x_{(p+1)\pi/2},\ x_{(p+2)\pi/2},\ x_{(p+3)\pi/2},\ x_{(p+4)\pi/2}\Big). \end{aligned} \tag{6.5}$$

The initial extracted features, $p = 0$, are extracted from the sliding window and are expressed as

$$\begin{aligned}\mathbf{x}_0 &= \Big(f_1(0),\ f_2(0),\ f_3(0),\ f_4(0),\ f_5(0)\Big) \\ &= \Big(x_0,\ x_{\pi/2},\ x_{\pi},\ x_{3\pi/2},\ x_{2\pi}\Big).\end{aligned} \tag{6.6}$$

It should be noted that the length of the sliding window in this example is set at Q=5. The position of the sliding window is incremented by 1 (equivalent shift of $\frac{\pi}{2}$) to evaluate a new range of observations in the time series (figure 6.2), which is expressed as

$$\begin{aligned}\mathbf{x}_1 &= \Big(f_1(1),\ f_2(1),\ f_3(1),\ f_4(1),\ f_5(1)\Big) \\ &= \Big(x_{\pi/2},\ x_{\pi},\ x_{3\pi/2},\ x_{2\pi},\ x_{5\pi/2}\Big).\end{aligned} \tag{6.7}$$

As the position is incremented, the five features extracted from the time series in set $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$ are presented to a clustering method. To understand the claim of Keogh [29], focus will only be placed on the first feature $f_1(p)$ without loss of generality. The feature extracted at point $f_1(p)$ for the sliding window at position p is expressed as

$$f_1(p) = x_{p\pi/2}. \tag{6.8}$$

Equation (6.8) is used to create a time series $\mathbf{f}_1$ for all the values of $f_1(p)$ for all positions p of the sliding window and is expressed as

$$\mathbf{f}_1 = \Big(x_0,\ x_{\pi/2},\ x_{\pi},\ \ldots x_{(\mathcal{I}-Q)\pi/2}\Big). \tag{6.9}$$

The values of the triply modulated cosine function is substituted into $\mathbf{f}_1$ as

$$\mathbf{f}_1 = \Big(\alpha_i,\ \mu_i,\ -\alpha_i,\ \mu_i, \alpha_i\ \ldots \alpha_i\Big). \tag{6.10}$$

This shows that inadvertly all the features are sequentially presented to every dimension of the feature vector. The fundamental problem becomes intuitive, as every feature dimension is sequentially attempting to learn the same thing. This is better illustrated by tabulating the set of features $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$. Table 6.1 shows what each feature point measures as a function of the sliding window increments. □

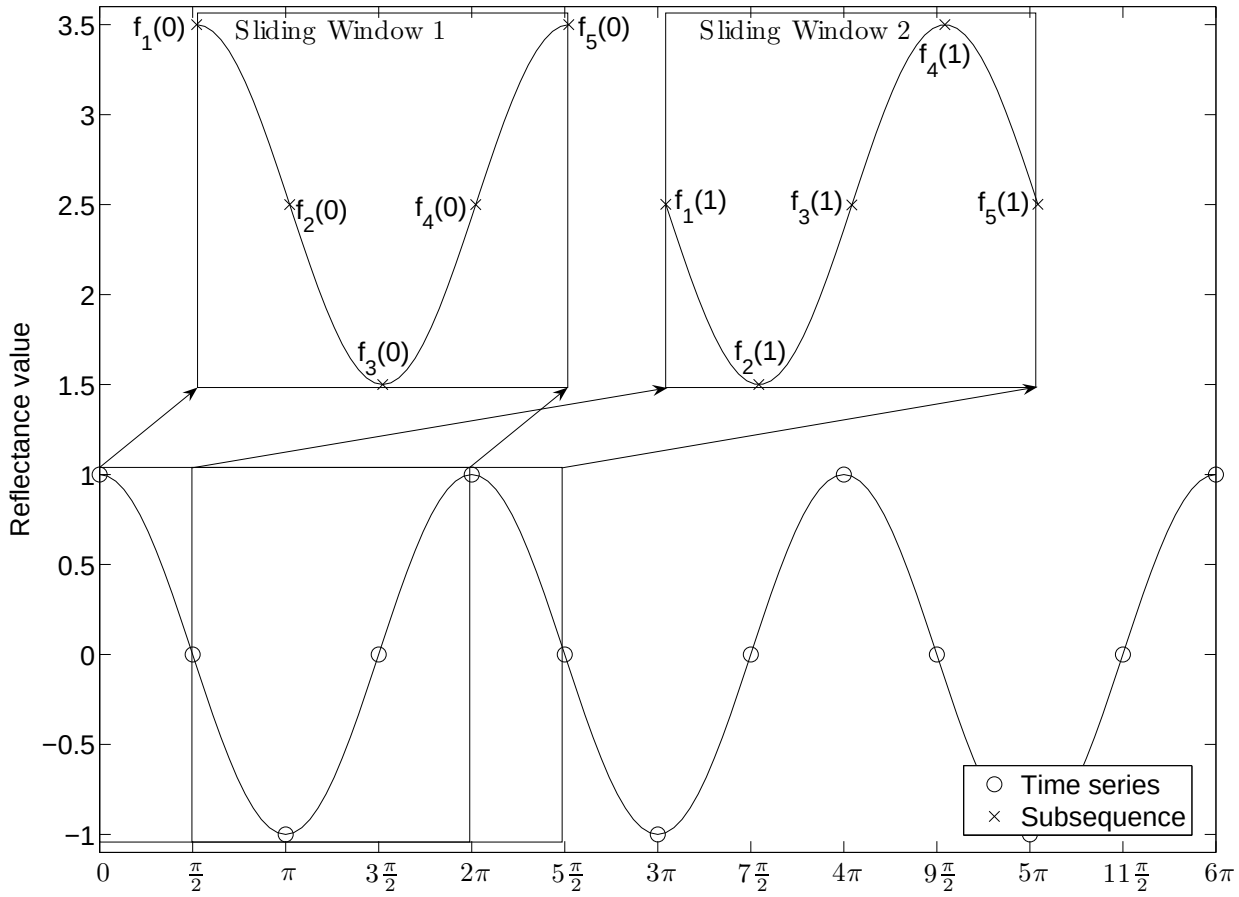

FIGURE 6.2: Two sets of five feature points $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$, are separated by a period of $\frac{\pi}{2}$, are shown to be extracted by two sliding windows.

Table 6.1: The sequence of features extracted as a function of the sliding window's position from figure 6.2.

Sliding window position	**Time increment**	**Feature points**				
		f_1	f_2	f_3	f_4	f_5
0	0	α_i	μ_i	-α_i	μ_i	α_i
1	$\frac{\pi}{2}$	μ_i	-α_i	μ_i	α_i	μ_i
2	π	-α_i	μ_i	α_i	μ_i	-α_i
3	$\frac{3\pi}{2}$	μ_i	α_i	μ_i	-α_i	μ_i
4	2π	α_i	μ_i	-α_i	μ_i	α_i

The intuition behind understanding this problem is to imagine an arbitrary data point somewhere in the time series which enters the sliding window and the contribution this data point makes to the overall mean of the sliding window. As the sliding window passes by, the data point first appears as the rightmost value in the window and then sequentially appears exactly once in every possible location within the sliding window. Thus all feature points will present the same information at different times and different dimensions to the clustering algorithm. This is equivalent to only presenting one data

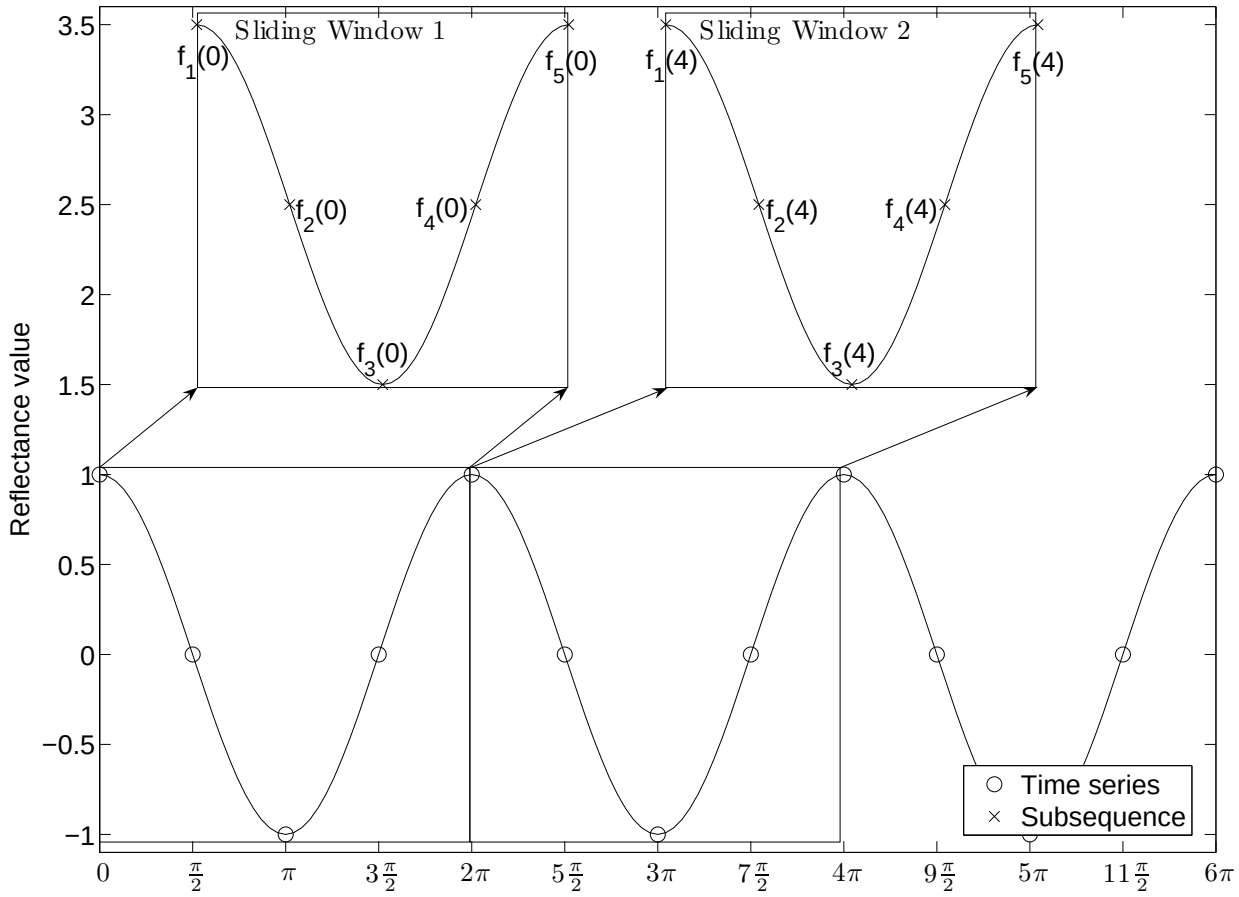

FIGURE 6.3: Two sets of five feature points $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$, are separated by a period of 2π, are shown to be extracted by two sliding windows.

point to a clustering algorithm and sequentially shifting through the time series.

Several ideas were formulated on how to create meaningful clusters [29]. The first idea was to increment the position of the sliding window by more than the length of the sliding window. This does not solve the problem, as the *subsequence clustering* becomes a *whole clustering* application. The second idea considered by Keogh and Lin [29] was to set the number of clusters much higher than the true number of clusters within the data set. Empirically this only worked if the number of clusters was set impractically high. The authors concluded that there is no simple solution to the problem of *subsequence clustering*.

Proposition 6.3.1 *A tentative solution was presented by Keogh and Lin [29] to find meaningful clusters using subsequence clustering. The example is in essence whole clustering, but it does emphasise an interesting property. The tentative solution proposes a single time series with a repetitive pattern, as shown in figure 6.3. The sliding window is shifted by exactly one period of the repetitive pattern within the time series. The new features are extracted and presented to the clustering algorithm. The solution becomes more intuitive if the features are tabulated in sequence of extraction.*

Table 6.2: The sequence of features extracted as a function of the sliding window's position from figure 6.3.

Sliding window position	**Time increment**	**Feature points**				
		f_1	f_2	f_3	f_4	f_5
0	0	α_i	μ_i	-α_i	μ_i	α_i
1	2π	α_i	μ_i	-α_i	μ_i	α_i
2	4π	α_i	μ_i	-α_i	μ_i	α_i
3	6π	α_i	μ_i	-α_i	μ_i	α_i
4	8π	α_i	μ_i	-α_i	μ_i	α_i

Table 6.2 now shows that each feature point is acquiring a single property of the time series. Through feature selection it becomes apparent that features f_3–f_5 can be discarded. This tentative solution provides meaningful clusters when the sliding window position p is incremented by the period of the repetitive pattern.

This however becomes a whole clustering *solution if the sliding window's position is incremented by more than its length. This results in analysing non-overlapping sliding windows.* □

Since remote sensing time series data have a strong periodic component due to the seasonal vegetation dynamics, the extracted sequential time series could potentially be processed to yield usable features. A feature extraction method is proposed in the next section that will reduce the feature space's dimensionality and removes the restriction of the tentative solution proposed in [29]. The removal of the restriction on the sliding window's position p will enable effective subsequence clustering that does not suffer from the afore-mentioned limitations.

6.4 MEANINGFUL CLUSTERING

In this section a method is shown that will create usable features from a subsequence $\mathbf{x}_p$ extracted from a MODIS MCD43A4 time series data set. The fixed acquisition rate of the MODIS product and the seasonality of the vegetation in the study area make for an annual periodic signal $\mathbf{x}$ that has a phase offset that is correlated with rainfall seasonality and vegetation phenology. The FFT [200] of $\mathbf{x}_p$ is computed, which decomposes the time sequence's values into components of different frequencies with phase offsets. This is often referred to as the frequency (Fourier) spectrum of the time series. Because the time series $\mathbf{x}_p$ is annually periodic, this would translate into frequency components in the frequency spectrum that have fixed positions with varying phase offsets. The varying phases limits the shifting of the sliding window's position p to exactly a periodic cycle [29], except if the clustering algorithm can cater for the varying phases.

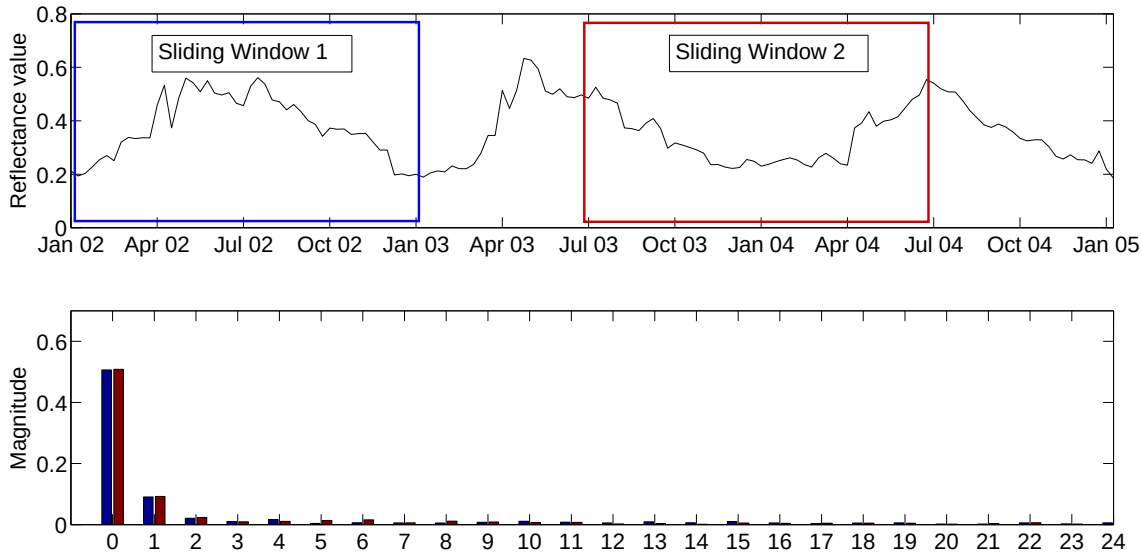

FIGURE 6.4: The feature components $X_p(f)$ extracted from two sliding windows at random positions using equation (6.11) yields similar features.

This limitation is addressed by computing the magnitude of all the FFT components, which removes all the phase offsets. This makes it possible to compensate for both the restrictive position p of the sliding window and the seasonality. This means that p, which is the position of the sliding window, does not have to be incremented by only a fixed annual period, but can be incremented by any natural number. The features for the clustering method are extracted from the sliding window $\mathbf{x}_p$ by the methodology discussed above, and are termed as the SFF $\mathcal{X}_p$. The SFF is computed as

$$\mathcal{X}_p = |\,\mathfrak{F}(\,\mathbf{x}_p\,)\,|, \tag{6.11}$$

where $\mathfrak{F}(\cdot)$ represents the Fourier transform. From the discussion above, a sliding window of any length can be applied to the MODIS time series and moved along the time axis at any rate as long as the feature extraction rule in equation (6.11) is applied. Figure 6.4 illustrates how the SFFs that are extracted using two different sliding window positions in time maintain their position in the feature space, even though the two sliding windows are arbitrarily positioned in time.

The seasonal attribute typically associated with MODIS time series and the slow temporal variation relative to the acquisition interval [15], makes the first few FFT components dominate the frequency spectrum. This reduces the number of features needed to represent the feature space and thus reduces the dimensionality, making clustering an even more feasible option [201].

The mean and annual FFT components from equation (6.11) were considered, as it was shown by Lhermitte [116] that considerable class separation can be achieved from these components. Many

FFT-based classification and segmentation methods consequently only consider a few FFT components [116, 202, 203].

6.5 CHANGE DETECTION METHOD USING THE SEASONAL FOURIER FEATURES

In this section the meaningful clustering approach discussed in section 6.4 is incorporated into a land cover change detection method. The change detection method operates on multiple spectral bands, as shown in figure 6.5.

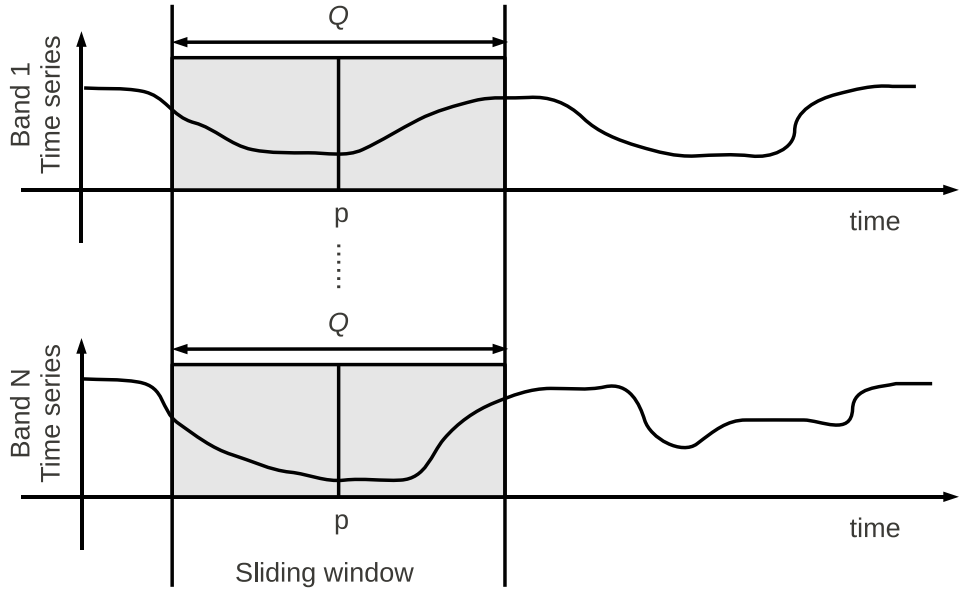

FIGURE 6.5: Temporal sliding window used to define a subsequence of the time series for classification and change detection.

The mean μ and annual α component of the SFF were considered from each of the MODIS spectral bands. These features are expressed using the same methodology discussed above as

$$\mathcal{X}_{bp} = |\, \mathfrak{F}_{b\mu}(\, \mathbf{x}_{bp}) \;\; \mathfrak{F}_{b\alpha}(\, \mathbf{x}_{bp}\,)\,|, \tag{6.12}$$

where $\mathfrak{F}_{b\mu}$ denotes the mean component extracted from the b^{th} spectral band's Fourier transform. The function $\mathfrak{F}_{b\alpha}$ denotes the annual component extracted from the b^{th} spectral band's Fourier transform. The subsequence $\mathbf{x}_{bp}$ is extracted from the b^{th} spectral band at position p.

This selection of frequency components reduces the number of features to represent the feature space and thus reduces the dimensionality. A feature vector is defined to encapsulate multiple spectral bands' SFF. The feature vector is defined as

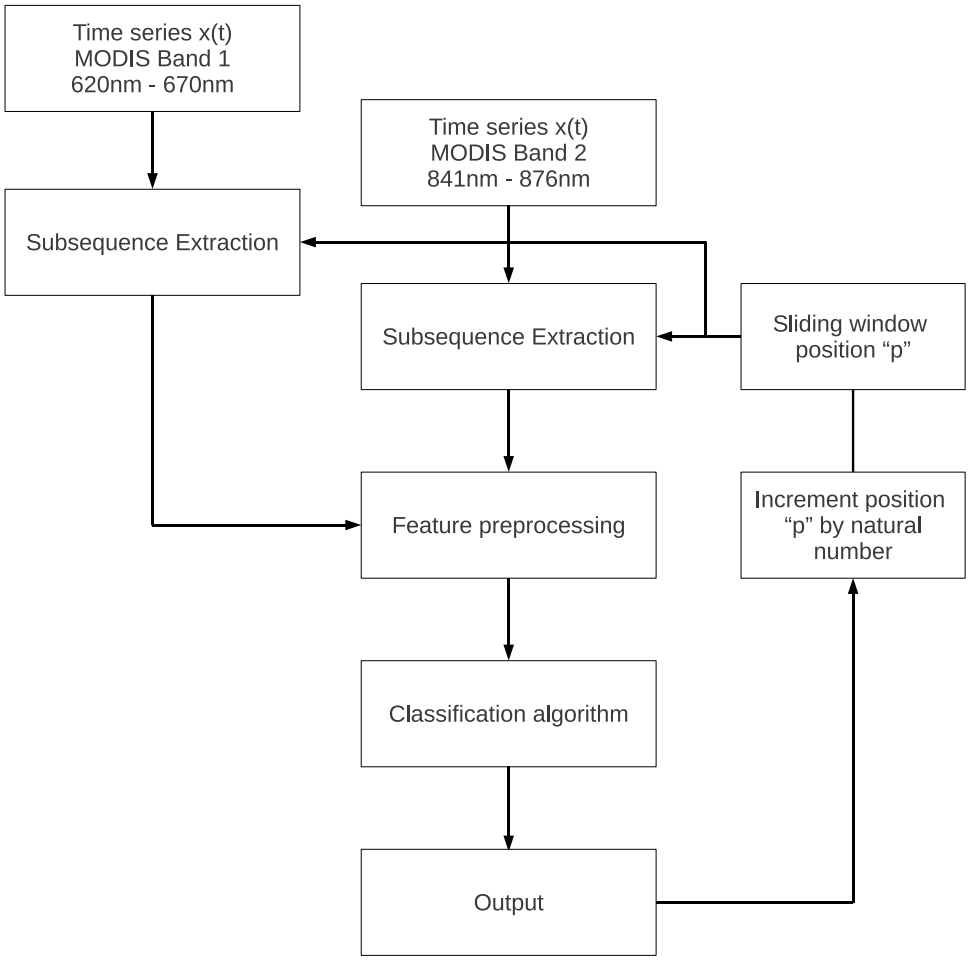

FIGURE 6.6: Subsequences of the time series extracted from the two spectral MODIS bands are processed for clustering and change detection.

$$\mathcal{X}_p^N = [\,\mathcal{X}_{1p}\;\; \mathcal{X}_{2p}\;\; \ldots \mathcal{X}_{Np}\,]. \tag{6.13}$$

Here N denotes the number of spectral bands, and p, $p \in [1, (\mathcal{I} - Q)]$, the position of the sliding window. The first feature vector is the NDVI time series (N=1), which is denoted by $\mathcal{X}_p^1$. This is where the NDVI is computed for $\mathcal{X}_{bp}$ in equation (6.1), which uses a combination of the first two spectral bands (RED and NIR spectral bands) of the MODIS instrument. The second feature vector is to use the first two spectral bands separately (N=2), which is denoted by $\mathcal{X}_p^2$. The last feature vector uses all seven spectral bands separately (N=7), which is denoted by $\mathcal{X}_p^7$.

These SFFs are processed by a machine learning algorithm to detect change. The processing chain for the two spectral bands feature vector $\mathcal{X}_p^2$ is shown as an illustration in figure 6.6. The outputs produced a time series of classifications for a given pixel as a function of the sliding window position p. Land cover change is defined then as the transition in class label of a pixel's time series from one class to another class, after which it remains in the newly assigned class for the remainder of the time

series.

6.6 SUMMARY

In this chapter a detailed overview was given of the pitfall of creating meaningless clusters. An example was presented to illustrate the real limitation of subsequence clustering, followed by a few tentative solutions proposed by Keogh and Lin [29] to solve this problem. Keogh and Lin admit that these solutions are not a fully worked out solution to the problem, but with further investigation a possible solution could be identified. In section 6.5, the SFF was proposed as a solution for a particular data set, which in this case was a time series that had inherent seasonal variations. The SFF will be one of the extracted features used in chapter 8 to detect land cover change.