Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 2614 – 2634
© TÜBİTAK
doi:10.3906/elk-1602-341

# A novel approach for extracting ideal exemplars by clustering for massive time-ordered datasets

Ömer Faruk ERTUĞRUL*
Department of Electrical and Electronics Engineering and Architecture, Faculty of Engineering, Batman University, Batman, Turkey

**Received:** 25.02.2016 • **Accepted/Published Online:** 02.10.2016 • **Final Version:** 30.07.2017

**Abstract:** The number and length of massive datasets have increased day by day and this yields more complex machine learning stages due to the high computational costs. To decrease the computational cost many methods were proposed in the literature such as data condensing, feature selection, and filtering. Although clustering methods are generally employed to divide samples into groups, another way of data condensing is by determining ideal exemplars (or prototypes), which can be used instead of the whole dataset. In this study, first the efficiency of traditional data condensing by clustering approach was confirmed according to obtained accuracies and condensing ratios in 9 different synthetic or real batch datasets. This approach was then improved to be employed in time-ordered datasets. In order to validate the proposed approach, 23 different real time-ordered datasets were used in experiments. Achieved mean RMSEs were 0.27 and 0.29 by employing the condensed (mean condensed ratio was 97.17%) and the whole datasets, respectively. Obtained results showed that higher accuracy rates and condensing ratios were achieved by the proposed approach.

**Key words:** Data condensing, prototype extracting, clustering, massive datasets, time-ordered datasets

## 1. Introduction

Technological improvements and cost reductions in measurement, communication, and storage devices caused an extraordinary increase in the number and volume of both batch and time-ordered datasets. Therefore, analyzing massive datasets in an efficient way is one of the major issues in machine learning nowadays. Many different methods were proposed to achieve lower computational costs without degrading the accuracy, such as feature selection, data condensing, determining generalized exemplars, and rule-based approaches [1–5]. Unfortunately, these methods are generally focused on batch datasets. Owing to the instrumentational improvements of logging systems, especially in sampling frequency, there is a high requirement for a data reduction approach in time-ordered datasets.

By clustering, samples that have similar properties are categorized into a subset [6]. Each clustering method is built on a specific criterion to divide samples into groups; in general, these similarities between instances are measured via distance calculation methods [7]. For example, k-means clustering, which is one of the most commonly employed clustering methods because of its effectivity [8–10], is based on dividing samples into k different groups in which the intracluster similarities are maximized and the intercluster similarities are minimized [6]. This is achieved by an iterative process for determining optimal clustering centers and in this iterative process similarities are calculated based on the square error criterion [8]. Clustering methods have

*Correspondence: omerfaruk.ertugrul@batman.edu.tr been employed in many research problems such as biomedical datasets [11], high-dimensional problems [12], and times signals [13]. In addition to clustering, these methods were also employed to categorize samples in order to reduce the length of the dataset and determine ideal exemplars as a category definition [14–16]. For instance, Karegowda et al. employed k-means clustering in order to reduce the sample size by determining irrelevant samples [17]. It was reported that employing ideal exemplars instead of the whole dataset may improve accuracy while at the same time reducing the computational cost and the communication and storage requirements [5,18]. Due to these facts, there is a large and growing literature on prototype selection and generation [15,16,19,20].

The significance of time-ordered datasets increases day by day due to the increase in the utilization of data loggers. The main motivation behind this study is to build a humanoid-based approach in order to determine exact ideal exemplars because humans always reduce complex and various stimuli from the environment and make decisions depending on the concentrated stimuli. It was reported that in order to reduce these complex stimuli, humans categorize stimuli or objects based on their intrasimilarity by determining ideal exemplars (a prototype) or extracting rules [21–24]. In machine learning, categorization is normally done by classifying or clustering depending on whether the class of each sample in the training dataset is known or not, respectively [6].

In this study, first, k-means, which is a clustering method, is employed to categorize the batch datasets (synthetic and real datasets) in order to make the reducing capabilities of traditional clustering by condensing approach clearer. Ideal exemplars, which are central tendencies of samples in each cluster, were extracted from each cluster as a category definition. After achieving acceptable accuracies with good condensing ratios by traditional condensing by clustering approach, the methodology is improved to be employed in time-ordered datasets. The proposed approach is formulated in such a way that it can forget old samples and extract prototypes, ideal exemplars, or condensed datasets from the samples in the memory. Therefore, the tendency of each cluster is changed or updated based on the order of the query, akin to human learning [25]. Twenty-three different time-ordered datasets are employed in order to evaluate and validate the proposed approach. The results achieved by an ANN trained with a condensed dataset (by the proposed approach) are compared with results obtained by an ANN trained with the whole dataset and condensed dataset (by traditional condensing by clustering approach). Obtained results show that the proposed approach can be employed in time-ordered datasets successfully in terms of both achieved accuracy and condensing ratio. The rest of the paper is organized as follows: Section 2 explains a brief overview of datasets. Section 3 describes the proposed approach and the methodology of experiments. Section 4 presents results and outcomes of the proposed method. Finally, Section 5 concludes the study.

## 2. Material

In the first part of the study, 9 different batch datasets (synthetic or real) were employed in order to make the traditional condensing by clustering approach clearer. In the second part of the paper, 3 different types of time-ordered datasets, which are economic indicators, mean sea level, and solar radiation datasets, were employed to confirm the proposed approach.

### 2.1. Batch Datasets

In order to confirm the traditional condensing by clustering approach, synthetic datasets, which have different statistical distributions [26] (as seen in Figure 1), were generated by prtools [27] and employed in experiments. In addition to these synthetic datasets, to increase the cogency some real benchmark datasets were also employed and the properties of all employed batch datasets are summarized in Table 1. It can be easily observed from Figure 1 and Table 1 that each employed synthetic dataset has a different statistical distribution.

**Table 1.** Properties of employed batch datasets.

| Name | Type | Performed task | Number of classes and features | Properties |
|---|---|---|---|---|
| Lithuanian | Synthetic | Classification | 2 / 2 | The samples that belong to each class are uniformly distributed along a sausage [27]. |
| Highleyman | Synthetic | Classification | 2 / 2 | The samples that belong to the first class are normally distributed over the x-axis and the other samples are normally distributed over y-axis [27]. |
| Banana Shaped | Synthetic | Classification | 2 / 2 | The samples in this dataset are distributed along banana shapes [27]. |
| Spherical | Synthetic | Classification | 2 / 2 | The samples that belong to each class are spherically Gaussian distributed but the mean of Class 1 is 4 times higher than the means of the other class [27]. |
| Multi-Class | Synthetic | Classification | 8 / 2 | This dataset is a collection of the first four datasets [27]. |
| Pima Indian Diabetes | Real | Classification | 2 / 8 | This dataset consists of clinical features of diabetics (261 samples) and nondiabetics (501 samples) [28,29]. |
| Hepatitis | Real | Classification | 2 / 19 | This dataset consists of clinical features of 80 hepatitis patients; 47 of these samples belong to living patients while the others belong to deceased patients [29,30]. |
| Approximate Sinc Function | Synthetic | Regression | - / 1 | This dataset consists of approximate Sinc values that were calculated by adding a random noise, which lies in the range of –0.2 to +0.2, to the Sinc (sinc(x) = $\frac{\sin(x)}{x}$) of 5000 samples uniformly distributed on the interval [–10, 10] [31]. |
| CASP | Real | Regression | - / 1 | This dataset consists of physicochemical properties of protein tertiary structures and it was generated by taking the first 1000 samples from the CASP 5-9 dataset [29]. |

### 2.2. Time-ordered datasets

Twenty-three time-ordered datasets that belong to 3 different groups were employed to validate the proposed approach. The first group comprises financial indicator datasets (end-of-day value), which are the stock index, Forex, financial futures, energy, and commodities datasets that were downloaded from investing.com and are summarized in Table 2. The second group contains mean sea level (MSL) datasets, which were downloaded from the Permanent Service for Mean Sea Level that collects, publishes, and analyzes MSL data from a global

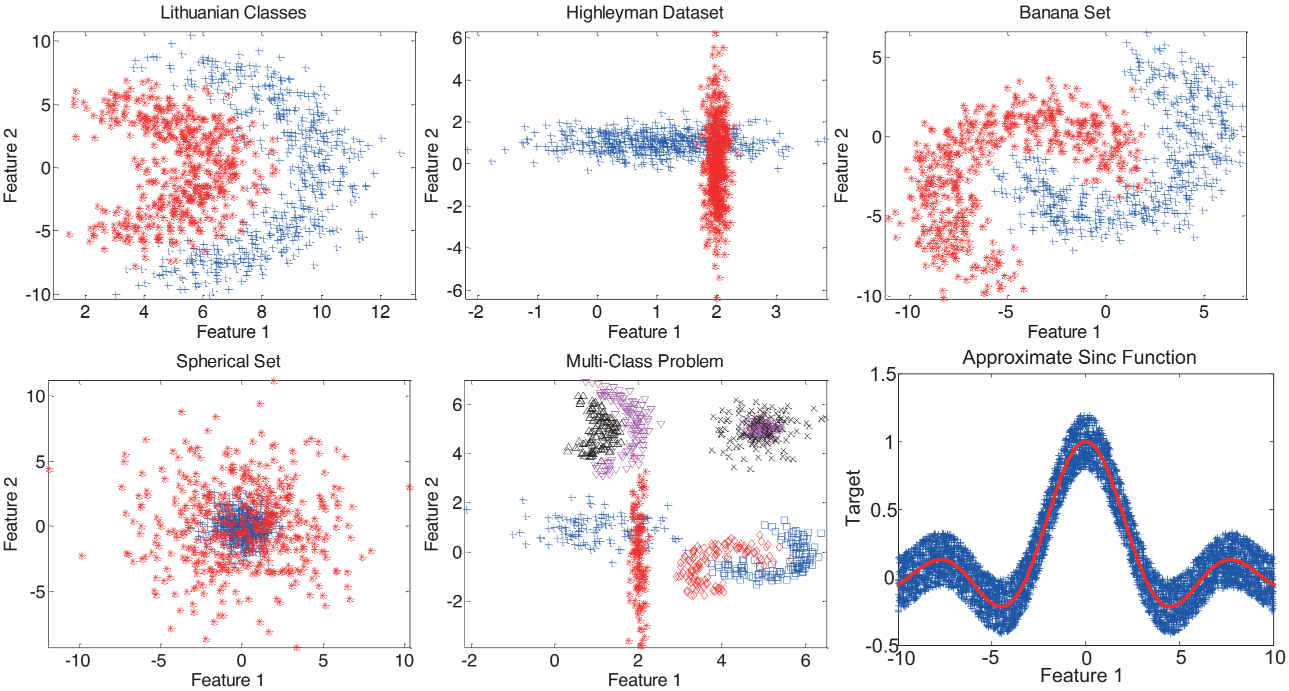

**Figure 1.** Employed synthetic batch datasets.

network database. The third group comprises solar radiation datasets that were downloaded from the US National Oceanic and Atmospheric Administration. The properties of the datasets that belong to the second and the third groups are given in Tables 3 and 4, respectively.

**Table 2.** Utilized economic indicators.

| Type | Periodicity | Dataset |
|---|---|---|
| Stock Index | Daily | S&P 500 Futures (USA, from 08.12.2005 to 27.02.2015), Dow 30 (USA, from 04.01.2007 to 27.02.2015), FTSE 100 (UK, from 07.01.2002 to 27.02.2015) |
| Forex | Daily | US Dollar Index (from 01.02.2007 to 27.02.2015), EUR/USD parity (from 01.01.2002 to 27.02.2015) |
| Financial futures | Daily | US 30Y T-Bond (from 08.12.2008 to 27.02.2015), Euro Bund (from 04.03.2008 to 27.02.2015) |
| Energy | Daily | Crude Oil (from 26.01.2006 to 27.02.2015), Natural Gas (from 26.01.2006 to 27.02.2015) |
| Commodities | Daily | Gold (from 23.01.2006 to 27.02.2015), Copper (from 13.04.2007 to 27.02.2015) |

As seen from Tables 2–4, the employed indexes, MSL stations, and solar stations were selected from different regions of the world in order to increase the cogency of the achieved results. The datasets that were summarized in these tables consist of two rows. One of them is the time of the record, which is the record day, the record month, and the record hour for datasets that are summarized in Tables 2, 3, and 4, respectively. The other one shows the recorded value, which is the index of the end of the day, monthly mean value, and total hourly value in the financial indicators, MSL, and solar radiation datasets, respectively.

**Table 3.** Properties of utilized MSL stations.

| ID | Location | Supplier | Periodicity | Metric data | Completeness | Source |
|---|---|---|---|---|---|---|
| 913 | 65.246233°S<br>64.257417°W | N.O.C. | Monthly | 1958–2013 | 97.8% | Antarctica |
| 2171 | 39.378472°N<br>31.168639°W | Instituto Hidrografico, Lisbon | Monthly | 2006–2013 | 100% | Portugal |
| 1391 | 27.083333°N<br>142.183333°E | Japan Meteorological Agency | Monthly | 1975–2013 | 99.1% | Japan |
| 2093 | 38.121411°N<br>13.371331°E | Ispra | Monthly | 2001–2013 | 100% | Italy |

**Figure 4.** The flowchart of the proposed approach for the time-ordered datasets.

| ID | Solar coordinate | Periodicity | Time zone | Location |
|---|---|---|---|---|
| 722255 | Latitude: 32.517°<br>Longitude: –84.95°<br>Elevation: 120 m | Hourly | –6 | Columbus Metropolitan Airport, GA, USA |
| 722700 | Latitude: 31.77°<br>Longitude: –106.5°<br>Elevation: 1186 m | Hourly | –7 | El Paso International Airport, TX, USA |
| 744860 | Latitude: 40.65°<br>Longitude: –73.8°<br>Elevation: 5 m | Hourly | –5 | New York John F. Kennedy International Airport, NY, USA |
| 911900 | Latitude: 20.9°<br>Longitude: –156.43°<br>Elevation: 16 m | Hourly | –10 | Kahului Airport, HI, USA |

In estimating these time-ordered datasets, the order of samples was used as input and the recorded values were estimated by using previous training samples by regression methods. Since each of these employed datasets belongs to a different field, each has its own periodicity based on its sampling period. For example, a sample was recorded for each hour in solar stations, while a sample was recorded for each month in the MSL dataset. The inputs were generated based on these periodicities because of the characteristics of the datasets, because each group of datasets was recorded based on a sampling frequency with which their values are associated [32–35]. Inputs of the daily datasets (see Table 2) were the order of the day of the week on which the sample was recorded (i.e. 1, 2, 3, 4, 5, 6, and 7 for samples recorded on Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday, respectively). Inputs of the monthly datasets (see Table 3) were the numbers of the months in which the samples were recorded (i.e. in the range of 1 to 12). Similarly, inputs of the hourly datasets (see Table 4) were the hours in which samples were recorded (i.e. inputs ranged from 1 to 24).

## 3. Method

### 3.1. Condensing by clustering in batch datasets

The flowchart of traditional condensing by clustering approach is shown in Figure 2 [14–16].

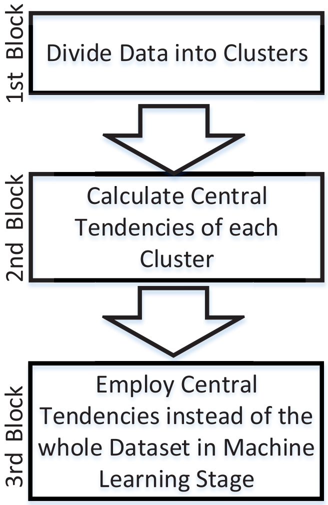
1st Block
Divide Data into Clusters
2nd Block
Calculate Central Tendencies of each Cluster
3rd Block
Employ Central Tendencies instead of the whole Dataset in Machine Learning Stage

**Figure 2.** The flowchart of traditional data condensing in batch datasets.

**1st Block:** In this block, samples in the training dataset are clustered by a clustering method such as the k-means clustering method. In classification problems, samples that belong to the same class are divided into clusters. On the other hand, in regression problems, the whole samples in the training dataset are clustered.

**2nd Block:** Central tendencies of each cluster are calculated in this block. Popular central tendency measures are the mode, median, and mean. The mean shows the center of gravity, which is also the balance point. The median represents the middle score and mode stands for the most frequent sample in the dataset [36]. The optimal central tendency measure can be selected based on whether the dataset contains error/noise or not. For example, the mean is more sensitive to noise, because it considers the whole dataset.

**3rd Block:** Central tendencies of each cluster, which is a condensed form of the whole dataset, are used instead of the dataset in the machine learning stage.

### 3.2. Proposed approach in time-ordered datasets

In the proposed approach, the dataset was employed due to the order of samples (events) for utilizing the knowledge gained from the order of the dataset, which may enhance the accuracy of both classification and regression. The proposed approach is summarized in Figure 3. As seen in Figure 3, for each query the proposed process was employed. The dataset was windowed for each query and the windowed part of the dataset (samples in the memory) was altered like in human learning (humans learn endlessly and by trial and error) [37]. The flowchart of the proposed approach is given in Figure 4 and described below.

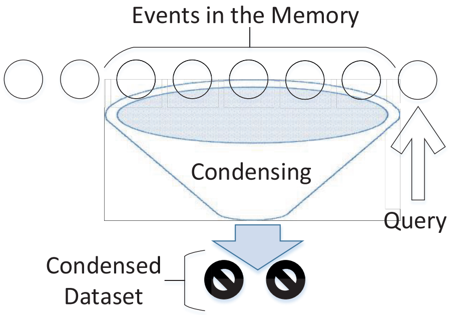

**Figure 3.** Evaluation of the proposed approach in time-ordered datasets.

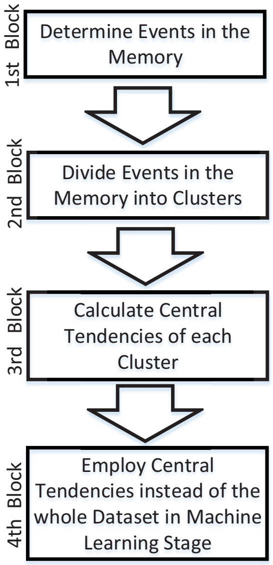
1st Block
Determine Events in the Memory
2nd Block
Divide Events in the Memory into Clusters
3rd Block
Calculate Central Tendencies of each Cluster
4th Block
Employ Central Tendencies instead of the whole Dataset in Machine Learning Stage

**Figure 4.** The flowchart of the proposed approach for the time-ordered datasets.

**1st Block:** In time-ordered datasets, the order of samples in the dataset may contain important information. Since time-ordered datasets are generally records of a natural phenomenon, the value of an event is related to its previous data. Based on this fact, some events/samples were picked out depending on the order of data. The selected samples the $n$th event was determined by the following equation.

$$\text{Selected Samples}(n) = \begin{cases} \{\text{dataset}(i) | 1 \leq i \leq \tau\}, & n \leq \tau \\ \{\text{dataset}(i) | n - \tau \leq i \leq n - 1\}, & n > \tau \end{cases} \tag{1}$$

Here, $\tau$ stands for the length of the selected data. These samples belong to the specifically sized part of the dataset, which is located before the query, standing for events in the memory. This process, which is akin to the human memory [38–40], can be visualized as a windowing process.

**2nd Block:** The events/samples in the memory were clustered by a clustering method such as k-means. In this study, the k-means clustering method was employed due to its effectivity.

**3rd Block:** The central tendencies of each cluster were calculated. These obtained tendencies stand for prototypes or generalized exemplars.

**4th Block:** The condensed dataset was employed to forecast or classify the query.

### 3.3. k-Means clustering

Clustering aims to divide a group of events/samples (a dataset) into subclasses/clusters in such a way that similarities of the intracluster are maximized while similarities of the intercluster are minimized [6]. This is achieved by determining the optimum cluster centers, which minimize the distortion of the samples in the clusters [41]. The process of optimizing cluster centers and members is done iteratively such that $k$ samples are selected arbitrarily as cluster centers and the cluster centers are updated until reaching optimum centers [6,17].

In k-means clustering, the unlabeled dataset is divided into $k$ clusters based on the similarity errors, which show the intradistortions based on the square error criterion. The square error is calculated by [6]:

$$E = \sum_{i=1}^{k} \sum_{x \epsilon C_i} \left\| x - \mu_i \right\|^2 \tag{2}$$

where $k$ is the number of clusters; $\mu_i$ is the cluster center of $C_i$, which stands for the cluster $i$; and $x$ represents the data. Although k-means clustering is a simple and popular clustering method, it suffers from a major drawback: there is no method to determine the optimum number of clusters into which the dataset must be divided [42].

### 3.4. Validation methods and metrics

In this study, two different cross-validation approaches were employed. The first was n-fold cross-validation. It was employed in batch-type datasets because in batch-type datasets the order of samples in the dataset is not important [43]. This procedure was employed as shown in Figure 5a. Each dataset was split into 10 subsets (here, $n$ was assigned as 10). In each of 10 epochs, a subset was employed as a test dataset while the others were used in training the employed machine learning method as seen in the 1st and 2nd epochs in Figure 5a. In this way, each sample in the dataset was estimated in tests. Finally, the mean of the obtained accuracies in all employed epochs was reported as overall accuracy [43,44]. In this way, achieved accuracy is less dependent on the order of samples. Therefore, this strategy cannot be employed in assessing the performance of a memory-based approach in time-ordered datasets. Instead of n-fold cross-validation, generally a Monte Carlo cross-validation method is employed in this type of learning process [45,46].

In Monte Carlo cross-validation, a group of samples is randomly selected as the training dataset while the others are selected as test samples and accuracy is calculated. This arbitrary selection process is repeated for $n$ epochs and the obtained mean accuracy is reported as the achieved accuracy [47]. Implementation of Monte Carlo cross-validation in this study is given in Figure 5b. Accuracies for each dataset were calculated by using 10 different epochs. The first 10% of the samples were used as a training dataset and the next 5% of the samples were estimated by employed machine learning methods based on training dataset (see the 1st epoch in Figure 5b). Later, the next 10% of samples, which means the first 20% of the samples, were added to the training dataset and similarly the next 5% of the samples were forecasted (see the 2nd epoch in Figure 5b). This process was repeated in the first 9 epochs. In the last epoch, 95% of the samples were used as the training dataset and the last part of the samples (the next 5%) were employed as test samples. Achieved accuracies were calculated simply by taking the mean of the obtained accuracies in 10 epochs.

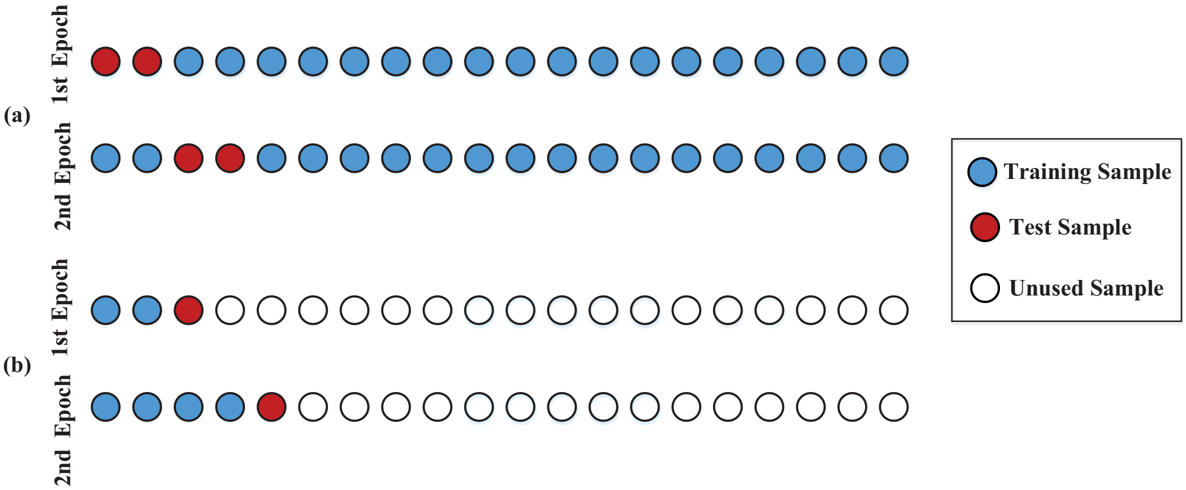

**Figure 5.** Employed cross-validation approaches: (a) n-fold cross-validation, (b) Monte Carlo cross-validation.

The simplest way of validating an approach, which is related to machine learning, is using the achieved success in benchmark datasets or different distributed synthetic datasets. In this study the employed validation metrics were accuracy (%) in classification and root mean square error (RMSE) in regression problems. They were calculated as follows.

$$\text{Accuracy}\ \ (\%) = \frac{\#\,\text{True classified data}}{\#\,\text{All data}} \times 100\% \tag{3}$$

$$RMSE = \sqrt{E\left[(f-y)^2\right]} \tag{4}$$

Here, $E$ is the expected value, $f$ is the true desired output, and $y$ is the forecasted output.

## 4. Results and discussion

### 4.1. Batch datasets

The obtained accuracy by the proposed approach is directly related to the number of clusters into which the dataset will be divided. Although there is a large and growing literature that reports successful results in employing clustering methods, clustering methods have a major drawback, which is about determining the optimum number of clusters [41,42]. As seen in the literature review, there is no exact way of determining the optimum number of clusters [42]. In order to make clear the relation between the number of clusters, which also shows the number of extracted samples (condensed data/ideal exemplars/prototypes), and obtained accuracies, the number of clusters was employed from 2 to 250 and condensed datasets were classified by kNN based on 10-fold cross-validation. Obtained accuracies are summarized in Table 5. Some of the values in Table 5 are missing, because $k$ cannot be assigned larger than the length of the dataset.

As seen in Table 5, no correlation was found between dataset length and the optimum number of the clusters, which well suits the literature findings [41,42]. The optimum number of the clusters may vary based on the properties of the dataset, such as the geometric distribution, statistical measures, and neighborhood measures [42,48]. In general, though, it can be reported that the increase in the number of clusters yields higher computational costs with lower condensing ratio and may also cause higher classification accuracy. As

**Table 5.** Obtained accuracies for different number of clusters.

| Dataset | | Number of clusters (condensed samples / ideal exemplars / prototypes) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 3 | 5 | 10 | 20 | 35 | 50 | 75 | 100 | 150 | 250 |
| Accuracy (%) | Lithuanian | 87.90 | 93.70 | 95.20 | 95.60 | 95.30 | 94.70 | 96.00 | 95.50 | 96.00 | 97.10 | 97.20 |
| | Highleyman | 50.00 | 85.00 | 82.80 | 80.80 | 83.40 | 89.30 | 88.90 | 90.40 | 90.00 | 93.30 | 93.90 |
| | Banana S. | 73.70 | 91.20 | 98.30 | 97.00 | 96.90 | 96.60 | 97.20 | 96.90 | 97.60 | 97.90 | 98.20 |
| | Spherical | 67.00 | 78.10 | 70.50 | 71.40 | 75.20 | 76.50 | 78.20 | 79.80 | 82.40 | 82.90 | 83.70 |
| | Multi-Class | 61.90 | 68.80 | 85.10 | 83.00 | 87.10 | 89.50 | 89.30 | - | - | - | - |
| | Diyabet | 65.06 | 64.94 | 66.10 | 67.01 | 67.27 | 67.79 | 69.61 | 70.00 | 70.13 | 68.70 | - |
| | Hepatitis | 51.25 | 47.50 | 57.50 | 48.75 | 55.00 | - | - | - | - | - | - |
| RMSE | Sinc | 5.74 | 5.74 | 5.74 | 5.75 | 5.76 | 5.76 | 5.77 | 5.78 | 5.79 | 5.81 | 5.88 |
| | CASP | 6.20 | 6.22 | 6.20 | 6.22 | 6.25 | 6.27 | 6.36 | 6.38 | 6.31 | 6.43 | 6.40 |

a consequence, $k$ must be determined as a balance between the accuracy and condensing ratio, and this can be simply achieved by trials (i.e. performing the classification process many times by employing different $k$ values). In order to make this process easier and to have fairer judgment, the number of clusters was assigned to $\sqrt{N/2}$, where $N$ is the number of instances in the dataset depending on the rule of thumb, due to its simplicity [49]. After assigning $k$ with a relation to the number of instances in the dataset (the length of the dataset), condensed datasets are illustrated in Figure 6.

As seen in Figures 1 and 6, the condensed dataset carries the fundamental characteristics of the entire dataset by utilizing a fewer number of samples. For instance, 21 ideal exemplars, which are means of the central tendencies of clusters, were employed instead of a dataset that consists of 1000 samples and in this case the condensing ratio was 97.90%. Achieved condensing ratios in the Lithuanian, Highleyman, Banana S., Spherical, Multi-Class, Diyabet, Hepatitis, Sinc, and CASP datasets are 97.90%, 97.90%, 97.90%, 97.90%, 97.90%, 97.66%, 97.50%, 99.06%, and 97.90%, respectively. In summary, larger datasets yield higher condensation ratios.

Both the whole and the condensed datasets were classified or estimated by popular machine learning methods. The nearest mean classifier (NMC), K-nearest neighbor (kNN), naive Bayes (NB), feedforward artificial neural network (ANN), support vector machine (SVM), and decision tree (DT) methods were employed for classification and linear regression (LR), kernel smooth regression (KSR), kNN regression, and Gaussian process regression (GPR) methods were used for regression. Obtained classification accuracies based on 10-fold cross-validation are presented in Table 6.

**Table 6.** Obtained classification accuracies (%) in batch datasets.

| Datasets | All samples | | | | | | Ideal exemplars | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NMC | kNN | NB | ANN | SVM | DT | NMC | kNN | NB | ANN | SVM | DT |
| Lithuanian | 84.15 | 96.55 | 92.95 | 96.10 | 83.15 | 92.75 | 78.10 | 94.60 | 87.30 | 94.80 | 84.20 | 92.50 |
| Highleyman | 75.20 | 91.75 | 90.20 | 90.45 | 86.55 | 90.15 | 75.90 | 89.30 | 90.60 | 88.80 | 83.60 | 85.70 |
| Banana S. | 81.35 | 98.85 | 93.65 | 98.55 | 86.45 | 91.85 | 80.60 | 97.30 | 90.10 | 97.30 | 85.30 | 95.20 |
| Spherical | 51.45 | 81.80 | 80.00 | 82.15 | 60.65 | 74.85 | 54.50 | 75.50 | 73.50 | 81.00 | 62.10 | 71.40 |
| Multi-Class | 70.55 | 90.80 | 80.80 | 74.95 | 51.65 | 88.25 | 70.80 | 88.50 | 81.90 | 74.90 | 51.00 | 82.40 |
| Diabetes | 62.96 | 75.07 | 75.00 | 76.24 | 77.21 | 66.02 | 63.51 | 67.92 | 69.22 | 70.78 | 76.88 | 67.79 |
| Hepatitis | 61.88 | 59.38 | 70.63 | 68.13 | 68.75 | 62.50 | 57.50 | 61.25 | 63.75 | 62.50 | 65.00 | 71.25 |

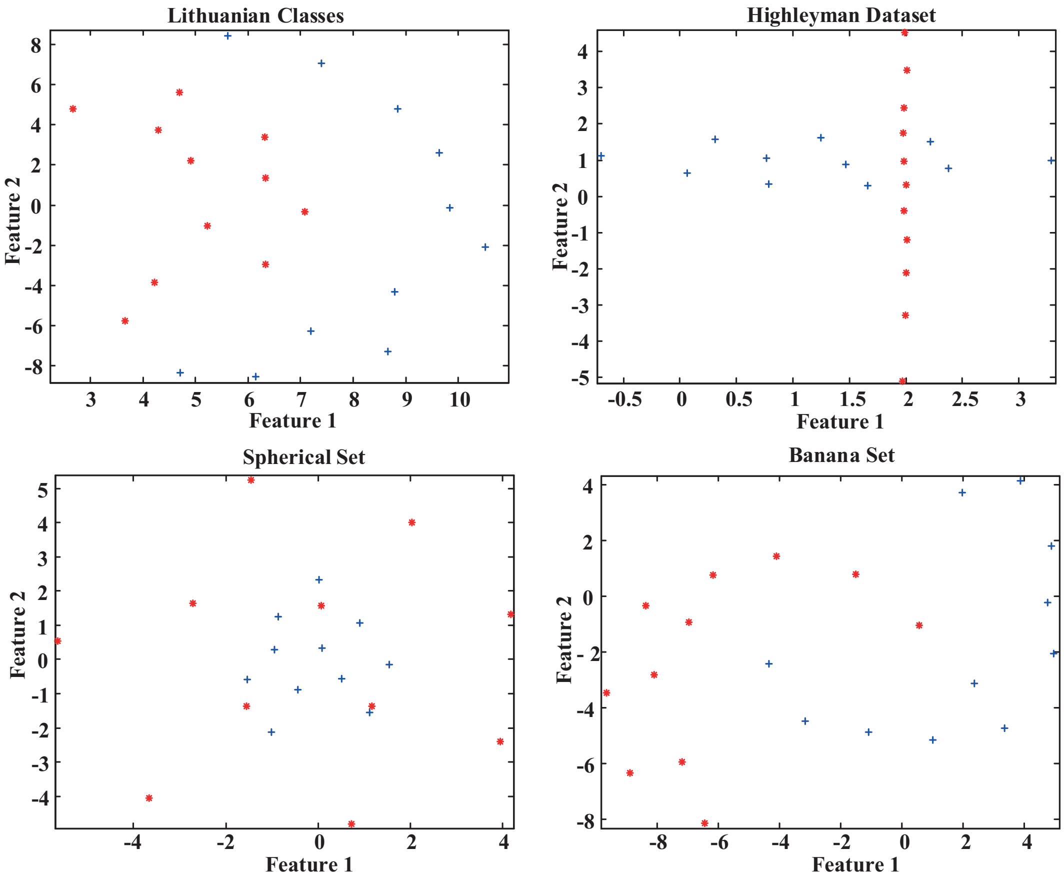

**Figure 6.** Extracted ideal exemplars in batch datasets.

As seen in Table 6, lower accuracies were obtained by employing the whole datasets (except the Hepatitis dataset), but the mean decrease in the accuracies of employed datasets (1.09) was in an acceptable range, especially when the mean condensing ratio (97.81%) was taken into account. Since samples belonging to these datasets were randomly generated based on the distributions, a fair comparison cannot be made between the achieved accuracies in this study and reported accuracies in the literature. However, similar accuracies were reported with synthetic datasets [26]. The Pima Indian Diabetes dataset was employed in many papers as a benchmark dataset and accuracies reported with ANNs trained by ELM, SVM, NB, and generalized behavioral learning methods were 77.57% [31], 76.50% [50], 64.60% [51], and 65.23% [52], respectively. Results obtained with the Diabetes dataset showed that higher accuracies by SVM and NB were achieved by employing a condensed dataset instead of employing the whole dataset. With Hepatitis datasets the reported accuracies obtained by NB and kNN were 65.7% and 69.6%, respectively [51], and higher accuracies were achieved by employing a condensed dataset. Additionally, obtained regression errors based on a 10-fold cross-validation scheme are summarized in Table 7.

As seen in Table 7, lower RMSEs were achieved with the Sinc dataset by utilizing the whole dataset instead of a condensed dataset, but with CASP dataset it was vice versa. In Bach datasets, higher accuracies were achieved by employing only the condensed dataset for Hepatitis and CASP datasets (as seen in Tables 6 and 7). These results may be because of the distribution of datasets or the employed methodology. Moreover, in order to investigate the relationship between the length of the dataset and the efficiency of the proposed

**Table 7.** Obtained RMSE in batch datasets.

| Datasets | All dataset | | | | | Ideal exemplar | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LR | KSR | kNN | GPR | ANN | LR | KSR | kNN | GPR | ANN |
| Sinc | 5.74 | 5.74 | 4.66 | 5.74 | 5.74 | 5.74 | 5.74 | 5.75 | 5.74 | 5.74 |
| CASP | 6.87 | 6.31 | 7.00 | 6.40 | 8.59 | 6.66 | 6.20 | 6.22 | 6.04 | 7.27 |

approach, the length of employed synthetic datasets was expanded from 1000 to 50,000 samples. These datasets were clustered depending on the rule of thumb and classified by kNN. Obtained classification accuracies (10-fold cross-validation) are tabulated in Table 8.

**Table 8.** Obtained accuracies for different number of samples.

| Number of samples | Number of clusters | Condensation ratio (%) | Datasets | | | |
|---|---|---|---|---|---|---|
| | | | Lithuanian | Highleyman | Banana S. | Spherical |
| 1000 | 21 | 97.90 | 94.60 | 89.30 | 97.30 | 75.50 |
| 1500 | 25 | 98.33 | 96.00 | 90.73 | 97.67 | 74.87 |
| 2000 | 30 | 98.50 | 96.30 | 88.85 | 97.03 | 75.45 |
| 2500 | 33 | 98.68 | 96.56 | 91.04 | 97.60 | 74.08 |
| 3000 | 36 | 98.80 | 96.63 | 91.01 | 97.67 | 75.83 |
| 4000 | 42 | 98.95 | 96.50 | 91.08 | 97.75 | 76.30 |
| 5000 | 47 | 99.06 | 96.84 | 92.74 | 97.90 | 76.90 |
| 10,000 | 67 | 99.33 | 96.80 | 92.72 | 97.86 | 77.28 |
| 20,000 | 94 | 99.53 | 96.84 | 92.76 | 98.05 | 77.61 |
| 50,000 | 150 | 99.70 | 96.95 | 93.19 | 98.31 | 78.31 |

As seen in Table 8, the increase in the length of the employed datasets yielded both higher accuracy and higher condensing ratio, as expected. These results showed that there is a correlation between the length of the dataset and achieved accuracies. This may be explained by the growth of the representative power of condensed datasets with enlargement of the dataset. Furthermore, used time and obtained P-values (t-test) of the employed datasets are summarized in Table 9. In the tests, kNN was employed based on 10-fold cross-validation. In this table, SL, ML, input, and output represent determining ideal exemplars, employing a machine learning method (here it is kNN), t-test between inputs of the whole dataset and inputs of extracted ideal exemplars, and t-test between outputs of the whole dataset and outputs of extracted ideal exemplars, respectively. As seen in Table 9, total used time (determining ideal exemplars and machine learning stage) is lower than the process time for the whole dataset. This required time may change based on the employed machine learning method, but in general, a machine learning method can be trained faster by using a smaller number of samples instead of a whole dataset [53–55]. In addition to the requirement of less time, the requirement of the memory of condensing by clustering method is lower than in traditional applications [1,2,6]. Furthermore, the obtained P-values showed that the extracted input and output values are higher than 0.05 (see Table 9). Obtained P-values showed that the extracted ideal exemplars came from the same distributions with the employed datasets.

### 4.2. Time-ordered datasets

The findings in batch datasets showed that tolerable accuracies were obtained by using condensed datasets instead of the whole dataset based on achieved condensing ratios. In this part of the study, this approach is improved to be employed in time signals in order to achieve higher accuracies. The proposed approach was

**Table 9.** Used time and obtained P-values.

| Datasets | Used time (s) | | | P-value (obtained by t-test) | |
|---|---|---|---|---|---|
| | ML | Proposed approach | | Input | Output |
| | | SL | ML | | |
| Lithuanian | 0.118 | 0.049 | 0.026 | 0.171 | 0.804 |
| Highleyman | 0.113 | 0.043 | 0.025 | 0.976 | 0.436 |
| Banana S. | 0.119 | 0.043 | 0.027 | 0.085 | 0.919 |
| Spherical | 0.111 | 0.040 | 0.025 | 0.363 | 0.778 |
| Multi-Class | 0.184 | 0.045 | 0.033 | 0.070 | 0.906 |
| Diabetes | 0.176 | 0.046 | 0.024 | 0.810 | 0.254 |
| Hepatitis | 0.122 | 0.039 | 0.027 | 0.345 | 0.419 |
| Sinc | 0.134 | 0.037 | 0.014 | 0.988 | 0.700 |
| CASP | 0.195 | 0.051 | 0.015 | 0.912 | 0.660 |

evaluated and validated via the time-ordered datasets described in Section 2.2. In these processes, sample order based on the periodicity of the assessed dataset was employed as an input while its value was utilized as output. An ANN trained by an extreme learning machine (ELM) was used in experiments due to its high generalization capability and extremely fast training stage [31]. The number of neurons and the transfer function in the hidden layer were assigned as 5 and triangular basis, respectively. To investigate the relation between the length of the dataset in the memory and the success of the proposed approach, different memory lengths were evaluated in Dow Jones 30 indexes and obtained RMSEs are listed in Table 10.

**Table 10.** Obtained accuracy (RMSE) for different training dataset ratio.

| Ratio of training dataset | Length of memory | Number of clusters | Condensing ratio (%) | $\text{ANN}^{proposed}$ |
|---|---|---|---|---|
| 5% | 102 | 7 | 99.66 | 0.1614 |
| 10% | 204 | 10 | 99.51 | 0.1859 |
| 20% | 408 | 14 | 99.31 | 0.1926 |
| 30% | 612 | 17 | 99.17 | 0.1924 |
| 50% | 1020 | 22 | 98.92 | 0.2001 |
| 75% | 1530 | 27 | 98.68 | 0.2231 |

As seen in Table 10, at lower memory sizes, achieved RMSE values are less than the RMSE obtained by the ANN trained with the whole dataset, which is 0.2054. However, the increase in memory size yields not only higher RMSEs but also lower condensing ratios. This is due to the fact that the larger the memory size reduces the knowledge gained from the data order. No correlation was found for the optimum memory length. It can be determined by experts depending on the features of the modeled system or by trials. Based on this fact, in the condensing procedure of the experiments, $\tau$ was assigned as 10% of the length of the employed dataset. Additionally, the number of clusters was assigned depending on the rule of thumb and the length of employed datasets, condensed datasets, and achieved condensing ratios for both condensing by clustering and the proposed approach are summarized in Table 11.

As seen in Table 11, higher condensing ratios were obtained by the proposed approach compared to the traditional condensing by clustering approach because, in the proposed approach, the condensed dataset was extracted from the samples in the memory instead of the whole dataset. Therefore, less storage capacity and lower computational cost were required in the proposed approach compared to the traditional condensing by

**Table 11.** Obtained condensing ratios for time-ordered datasets.

| Dataset type | | Dataset | Data length | Condensed by clustering (traditional approach) | | Condensed by the proposed approach | |
|---|---|---|---|---|---|---|---|
| | | | | Condensed data length | Condensing ratio (%) | Condensed data length | Condensing ratio (%) |
| Daily Economic Indicators | Stock Index | Dow 30 | 2040 | 32 | 98.43 | 10 | 99.51 |
| | | S&P 500 F. | 2344 | 34 | 98.55 | 10 | 99.57 |
| | | FTSE 100 | 3321 | 41 | 98.77 | 12 | 99.64 |
| | Forex | US Dollar I. | 2089 | 32 | 98.47 | 10 | 99.52 |
| | | EUR/USD | 4105 | 45 | 98.9 | 14 | 99.66 |
| | Financial Futures | US 30Y T-B. | 1613 | 28 | 98.26 | 8 | 99.50 |
| | | Euro Bund | 1783 | 30 | 98.32 | 9 | 99.50 |
| | Energy | Crude Oil | 2306 | 34 | 98.53 | 10 | 99.57 |
| | | Natural Gas | 2306 | 34 | 98.53 | 10 | 99.57 |
| | Commodities | Gold | 2241 | 33 | 98.53 | 10 | 99.55 |
| | | Copper | 2304 | 34 | 98.52 | 10 | 99.57 |
| Monthly SL | Sea Level Station ID | 913 | 672 | 18 | 97.32 | 5 | 99.26 |
| | | 2171 | 96 | 7 | 92.71 | 2 | 97.92 |
| | | 1391 | 456 | 15 | 96.71 | 4 | 99.12 |
| | | 2093 | 156 | 9 | 94.23 | 2 | 98.72 |
| Hourly SR | Solar Station ID | 722255 | 70129 | 187 | 99.73 | 59 | 99.92 |
| | | 722700 | 70129 | 187 | 99.73 | 59 | 99.92 |
| | | 744860 | 70129 | 187 | 99.73 | 59 | 99.92 |
| | | 911900 | 70129 | 187 | 99.73 | 59 | 99.92 |

clustering approach [53–55]. To assess the representative power of the extracted generalized exemplars from the employed dataset, unpaired t-tests were employed and achieved P-values are summarized in Table 12.

As seen in Table 12, P-values obtained by the proposed approach, which were higher than 0.05 for each case, showed that the extracted generalized exemplars are related to the employed datasets. Obtained P-values from the outputs of extracted exemplars from MSL datasets were higher than those of the other employed time-ordered datasets. This may be because of the characteristics of this type of dataset. In MSL datasets, there are a small trend and a small change based on the seasons. Similarly, fluctuations in some financial indicators such as EUR/USD, US 30Y T-B, and Euro Bund were lower than the other employed financial indicators and the obtained P-values for these indicators were also higher than obtained P-values by ANN$^{clustering}$.

In order to validate the success of the proposed approach, obtained mean accuracies of 10 epochs (Monte Carlo cross-validation) by using the whole dataset (ANN$^{whole}$), samples before the query (the length of previous samples is 10% of the length of the whole dataset, ANN$^{10\%}$), previous samples before the query with a sliding window method (ANN$^{window}$), extracted samples by autoregressive (AR) model (ANN$^{AR}$), condensed dataset by traditional condensing by clustering approach (ANN$^{clustering}$), and the condensed dataset by the proposed approach (ANN$^{proposed}$) are tabulated in Table 13. Note that ANN$^{window}$ was employed based on a sliding window technique in which the outputs of the previous instances $[t-m \dots t-1]$ were used as inputs to estimate

**Table 12.** Obtained P-values (t-test).

| Dataset type | | Dataset | ANN$^{10\%}$ | | ANN$^{clustering}$ | | ANN$^{proposed}$ | |
|---|---|---|---|---|---|---|---|---|
| | | | Input | Output | Input | Output | Input | Output |
| Daily Economic Indicators | Stock Index | Dow 30 | 0.967 | 0.932 | 0.029 | 0.048 | 0.321 | 0.261 |
| | | S&P 500 F. | 0.974 | 0.943 | 0.167 | 0.203 | 0.281 | 0.372 |
| | | FTSE 100 | 0.978 | 0.955 | 0.151 | 0.112 | 0.285 | 0.266 |
| | Forex | US Dollar I. | 0.961 | 0.959 | 0.318 | 0.056 | 0.307 | 0.237 |
| | | EUR/USD | 0.984 | 0.953 | 0.479 | 0.000 | 0.410 | 0.447 |
| | Financial Futures | US 30Y T-B. | 0.975 | 0.957 | 0.172 | 0.003 | 0.213 | 0.486 |
| | | Euro Bund | 0.964 | 0.956 | 0.059 | 0.000 | 0.470 | 0.694 |
| | Energy | Crude Oil | 0.974 | 0.951 | 0.161 | 0.016 | 0.536 | 0.224 |
| | | Natural Gas | 0.974 | 0.948 | 0.606 | 0.043 | 0.309 | 0.253 |
| | Commodities | Gold | 0.968 | 0.960 | 0.210 | 0.000 | 0.480 | 0.244 |
| | | Copper | 0.969 | 0.944 | 0.308 | 0.011 | 0.477 | 0.245 |
| Monthly SL | Sea Level Station ID | 913 | 0.953 | 0.980 | 0.664 | 0.235 | 0.666 | 0.906 |
| | | 2171 | 0.794 | 0.805 | 0.728 | 0.455 | 0.754 | 0.846 |
| | | 1391 | 0.914 | 0.893 | 0.638 | 0.375 | 0.858 | 0.925 |
| | | 2093 | 0.924 | 0.955 | 0.646 | 0.611 | 0.827 | 0.868 |
| Hourly SR | Solar Station ID | 722255 | 0.993 | 0.993 | 0.001 | 0.165 | 0.273 | 0.303 |
| | | 722700 | 0.993 | 0.993 | 0.000 | 0.000 | 0.349 | 0.268 |
| | | 744860 | 0.993 | 0.993 | 0.208 | 0.074 | 0.210 | 0.300 |
| | | 911900 | 0.993 | 0.993 | 0.120 | 0.002 | 0.234 | 0.201 |

the output at instant $t$ [56]. Here $m$ was assigned as two times the periodicity of the datasets given in Tables 2, 3, and 4. Additionally, for each dataset the order of the model in the AR method was determined by trials (i.e. changing the order of the AR model from 2 to 15) and obtained optimal orders ($n$) of the AR model in each dataset are also reported in Table 13.

It is obvious in Table 13 that the highest accuracies (i.e. lowest RMSEs) were achieved by ANN$^{proposed}$ compared to the other employed methods. Although acceptable accuracies were achieved by ANN$^{AR}$ in estimating financial indicators, obtained accuracies by ANN$^{AR}$ in SR datasets were low. This may be because of the characteristics of these datasets and the modeling procedure of AR. One of the interesting results from this table is that higher accuracies can be obtained by employing a memory-based method (ANN$^{10\%}$, ANN$^{window}$, and ANN$^{proposed}$) instead using the whole datasets (ANN$^{whole}$ and ANN$^{clustering}$). Additionally, the achieved mean accuracies by ANN$^{whole}$ were higher than the obtained results by ANN$^{clustering}$, but lower than the obtained accuracies by ANN$^{10\%}$ and ANN$^{window}$. The reason for this may be the change of the samples in the memory for each query. Therefore, condensed ideal exemplars (prototypes) were changed for each sample, similar to human learning or the object-recognizing methodology of humans. A human does not learn or memorize all things in his environment simultaneously [37]. Instead, a human categorizes everything; this categorization goes up to grouping the objects in classes and only the most representative ones are memorized [40]. These prototypes are changed when a new group is formed or a more representative prototype is recognized, which means they change by time. Humans only check an object with these prototypes and make decisions [37,40].

**Table 13.** Obtained accuracies (RMSE) for time-ordered datasets.

| Dataset type | | Dataset | ANN$^{whole}$ | ANN$^{10\%}$ | ANN$^{window}$ | AR method | | ANN$^{clustering}$ | ANN$^{proposed}$ |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | n | ANN$^{AR}$ | | |
| Daily Economic Indicators | Stock Index | Dow 30 | 0.2054 | 0.1843 | 0.1957 | 4 | 0.2173 | 0.2650 | 0.1859 |
| | | S&P 500 F. | 0.2175 | 0.2062 | 0.2063 | 9 | 0.1961 | 0.2666 | 0.1812 |
| | | FTSE 100 | 0.2377 | 0.2078 | 0.2045 | 3 | 0.2712 | 0.2910 | 0.1991 |
| | Forex | US Dollar I. | 0.1848 | 0. 1704 | 0. 1886 | 8 | 0.2128 | 0.1934 | 0.1827 |
| | | EUR/USD | 0.2150 | 0. 2014 | 0. 2075 | 7 | 0.1988 | 0.2566 | 0.1979 |
| | Financial Futures | US 30Y T-B. | 0.1976 | 0.1813 | 0.1998 | 7 | 0.2115 | 0.2622 | 0.1926 |
| | | Euro Bund | 0.2422 | 0.2353 | 0.2324 | 10 | 0.2362 | 0.2866 | 0.2220 |
| | Energy | Crude Oil | 0.2099 | 0.1940 | 0.2041 | 2 | 0.2154 | 0.2299 | 0.2019 |
| | | Natural Gas | 0.1718 | 0.1575 | 0.1432 | 5 | 0.1455 | 0.2025 | 0.1263 |
| | Commodities | Gold | 0.2852 | 0.2406 | 0.2508 | 10 | 0.2463 | 0.3007 | 0.2488 |
| | | Copper | 0.2285 | 0.2022 | 0.2254 | 2 | 0.2450 | 0.2945 | 0.2164 |
| Monthly SL | Sea Level Station ID | 913 | 0.4382 | 0.4348 | 0.4234 | 14 | 0.4520 | 0.6164 | 0.4229 |
| | | 2171 | 0.4372 | 0.4386 | 0.4298 | 4 | 0.4566 | 0.5834 | 0.4282 |
| | | 1391 | 0.4557 | 0.4466 | 0.4467 | 12 | 0.4565 | 0.5584 | 0.4401 |
| | | 2093 | 0.4442 | 0.4443 | 0.4221 | 7 | 0.4817 | 0.6607 | 0.4156 |
| Hourly SR | Solar Station ID | 722255 | 0.2453 | 0.2301 | 0.2306 | 10 | 0.4303 | 0.3836 | 0.2317 |
| | | 722700 | 0.2640 | 0.2535 | 0.2513 | 3 | 0.4785 | 0.3717 | 0.2485 |
| | | 744860 | 0.2493 | 0.2398 | 0.2410 | 8 | 0.4670 | 0.3696 | 0.2457 |
| | | 911900 | 0.2366 | 0.2376 | 0.2312 | 4 | 0.4455 | 0.3965 | 0.2292 |

**Table 14.** Process time (s).

| Dataset type | | Dataset | ANN$^{whole}$ | ANN$^{10\%}$ | ANN$^{window}$ | ANN$^{AR}$ | | ANN$^{clustering}$ | | ANN$^{proposed}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | SL | ML | SL | ML | SL | ML |
| Daily Economic Indicators | Stock Index | Dow 30 | 0.008 | 0.007 | 0.010 | 0.379 | 0.013 | 0.034 | 0.007 | 0.029 | 0.006 |
| | | S&P 500 F. | 0.010 | 0.009 | 0.011 | 0.349 | 0.013 | 0.027 | 0.007 | 0.029 | 0.007 |
| | | FTSE 100 | 0.009 | 0.009 | 0.009 | 0.366 | 0.013 | 0.026 | 0.007 | 0.031 | 0.007 |
| | Forex | US Dollar I. | 0.009 | 0.009 | 0.012 | 0.360 | 0.014 | 0.036 | 0.009 | 0.030 | 0.006 |
| | | EUR/USD | 0.009 | 0.009 | 0.010 | 0.362 | 0.013 | 0.028 | 0.008 | 0.031 | 0.006 |
| | Financial Futures | US 30Y T-B | 0.009 | 0.009 | 0.010 | 0.356 | 0.013 | 0.034 | 0.007 | 0.030 | 0.006 |
| | | Euro Bund | 0.010 | 0.009 | 0.010 | 0.366 | 0.013 | 0.034 | 0.008 | 0.032 | 0.006 |
| | Energy | Crude Oil | 0.009 | 0.010 | 0.011 | 0.366 | 0.014 | 0.027 | 0.008 | 0.031 | 0.007 |
| | | Natural Gas | 0.010 | 0.009 | 0.009 | 0.371 | 0.014 | 0.038 | 0.007 | 0.031 | 0.006 |
| | Commodities | Gold | 0.010 | 0.010 | 0.010 | 0.364 | 0.016 | 0.035 | 0.008 | 0.032 | 0.007 |
| | | Copper | 0.009 | 0.008 | 0.011 | 0.320 | 0.007 | 0.034 | 0.007 | 0.030 | 0.006 |
| Monthly SL | Sea Level Station ID | 913 | 0.009 | 0.009 | 0.010 | 0.352 | 0.013 | 0.034 | 0.008 | 0.029 | 0.006 |
| | | 2171 | 0.009 | 0.008 | 0.009 | 0.324 | 0.012 | 0.032 | 0.006 | 0.028 | 0.005 |
| | | 1391 | 0.009 | 0.008 | 0.009 | 0.328 | 0.013 | 0.033 | 0.007 | 0.028 | 0.006 |
| | | 2093 | 0.008 | 0.008 | 0.009 | 0.321 | 0.013 | 0.033 | 0.007 | 0.029 | 0.006 |
| Hourly SR | Solar Station ID | 722255 | 0.012 | 0.010 | 0.010 | 0.870 | 0.01 | 0.026 | 0.010 | 0.030 | 0.006 |
| | | 722700 | 0.015 | 0.011 | 0.012 | 0.778 | 0.013 | 0.028 | 0.008 | 0.032 | 0.007 |
| | | 744860 | 0.019 | 0.013 | 0.014 | 0.869 | 0.016 | 0.031 | 0.011 | 0.042 | 0.006 |
| | | 911900 | 0.018 | 0.012 | 0.016 | 0.788 | 0.014 | 0.027 | 0.008 | 0.035 | 0.008 |

In order to validate that this idea, the order of the samples in the Dow Jones 30 Index dataset was arbitrarily changed and obtained RMSEs by ANN$^{whole}$, ANN$^{10\%}$, ANN$^{window}$, ANN$^{AR}$, ANN$^{clustering}$, and ANN$^{proposed}$ were 0.2145, 0.2187, 0.2257, 0.2154, 0.2173, 0.2273, and 0.2100, respectively. As seen from these results and Table 13, lower RMSE values were obtained by using datasets in their natural order. Although the higher P-value obtained in the t-test shows that the higher probabilities of both datasets (original and extracted datasets) are a part of the same dataset, achieved RMSE values by the proposed approach were higher for the datasets for which high P-values were obtained (see Table 12). Additionally, in order to investigate the computational costs of employed methods, mean process times for each stage are reported in Table 14. It is obvious from Table 14 that the SL and ML stages by the proposed approach took much more time than the process time of ANN$^{whole}$ and ANN$^{10\%}$. This is because of the extremely fast training stage of the ELM [31]. Additionally, the proposed approach was faster than the other employed condensing approaches, which are ANN$^{AR}$ and ANN$^{clustering}$.

The obtained results in this study showed that higher or tolerable accuracies can be obtained by using lower numbers of samples, which were extracted by the proposed approach, in time-ordered signals. This result suits the literature in that higher accuracies can be obtained by employing ideal exemplars instead of the whole dataset [5,18]. Achieving higher or similar accuracies (by using lower numbers of samples in the training dataset) compared to popular time series analysis approaches showed that the proposed approach has high potential to be employed in analyzing time-ordered datasets, as utilizing fewer samples not only decreases the requirement of storage capacity but also may decrease the computational cost based on the employed machine learning method [5,18]. As a consequence, in this study, the contributions that were gained by the traditional condensing by clustering method that has been employed successfully in batch datasets were transferred to be used with time-ordered datasets.

## 5. Conclusion

Due to technological improvements, the functionality and usability of data loggers have increased, and based on this fact there has been a significant increase in the number and length of massive time-ordered datasets. Therefore, there is a requirement for a methodology that can condense massive time-ordered datasets without losing the knowledge gained from the data orders. In this study, the condensing via clustering methodology was first validated by utilizing 9 different batch datasets. Slightly lower but tolerable accuracies depending on the achieved condensing ratios were achieved by condensed datasets, which were extracted by clustering. Later, a novel concept, memory, was added to the employed approach in order to employ it for time-ordered datasets. Twenty-three different time-ordered datasets were employed to validate the proposed approach. Higher accuracies were achieved by employing a condensed dataset compared to employing the whole dataset, previous samples, samples modeled by AR, and condensed datasets by the traditional clustering approach. The results showed that the proposed approach can be successfully employed as a data condensing method for time-ordered datasets.

## References

[1] Dash M, Liu H. Feature selection for classification. Intell Data Anal 1997; 1: 131-156.

[2] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003; 3: 1157-1182.

[3] Ladha L, Deepa T. Feature selection methods and algorithms. Int J Comput Sci Eng 2011; 3: 1787-1797.

[4] Datta RP, Saha S. An Empirical Comparison of Rule Based Classification Techniques in Medical Databases. Working Paper IT-11-07. New Delhi, India: Indian Institute of Foreign Trade, 2011.

[5] Wilson DR, Martinez TR. Reduction techniques for instance-based learning algorithms. Mach Learn 2000; 38: 257-286.

[6] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Comput Surv 1999; 31: 264-323.

[7] Balcan MF, Blum A, Vempala S. Clustering via similarity functions: theoretical foundations and algorithms. In: 40th ACM Symposium on Theory of Computing Conference; 17–20 May 2008; Victoria, Canada. New York, NY, USA: ACM. pp. 1-42.

[8] Xu R, Wunsch D. Survey of clustering algorithms. IEEE T Neural Networ 2005; 16: 645-678.

[9] Likas A, Vlassis N, Verbeek JJ. The global k-means clustering algorithm. Pattern Recogn 2003; 36: 451-461.

[10] Li Y, Wu H. A clustering method based on K-means algorithm. Phys Procedia 2012; 25: 1104-1109.

[11] Xu R, Wunsch DC. Clustering algorithms in biomedical research: a review. IEEE Rev Biomed Eng 2010; 3: 120-154.

[12] Bouveyron C, Brunet-Saumard C. Model-based clustering of high-dimensional data: a review. Comput Stat Data Anal 2014; 71: 52-78.

[13] Liao TW. Clustering of time series data—a survey. Pattern Recogn 2005; 38: 1857-1874.

[14] Napoleon D, Pavalakodi S. A new method for dimensionality reduction using K-means clustering algorithm for high dimensional data set. Int J Comput Appl 2011; 13: 41-46.

[15] Ougiaroglou S, Evangelidis G. Fast and accurate k-nearest neighbor classification using prototype selection by clustering. In: 16th Panhellenic Conference on Informatics; 5–7 October 2012; Piraeus, Greece. pp. 168-173.

[16] Olvera-López JA, Carrasco-Ochoa JA, Martínez-Trinidad JF. A new fast prototype selection method based on clustering. Pattern Anal Appl 2010; 13: 131-141.

[17] Karegowda AG, Jayaram MA, Manjunath AS. Cascading K-means clustering and K-nearest neighbor classifier for categorization of diabetic patients. Int J Eng Adv Tech 2012; 1: 147-151.

[18] García S, Derrac J, Luengo J, Herrera F. A first approach to nearest hyperrectangle selection by evolutionary algorithms. In: Ninth International Conference on Intelligent Systems Design and Applications; 30 November–2 December 2009; Pisa, Italy. New York, NY, USA: IEEE. pp. 517-522.

[19] Gadodiya SV, Chandak MB. prototype selection algorithms for kNN classifier: a survey. Int J Adv Res Comp Comm Eng 2013; 2: 4829-4832.

[20] Triguero I, Derrac J, Garcia S, Herrera F. A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE T Syst Man Cy C 2012; 42: 86-100.

[21] Ashby FG, Maddox WT. Relation between prototype, exemplar and decision bound models of categorization. J Math Psychol 1993; 37: 372-400.

[22] Maddox WT, Ashby FG. Comparing decision bound and exemplar models of categorization. Percept Psychophys 1993; 53: 49-70.

[23] Medin DL, Schaffer MM. Context theory of classification learning. Psychol Rev 1978; 85: 207-238.

[24] Medin DL, Ross BH, Markman AB. Cognitive Psychology. 4th ed. New York, NY, USA: Wiley, 2005.

[25] Choromanska A, Monteleoni C. Online clustering with experts. In: International Conference on Artificial Intelligence and Statistics; 21–23 April 2012; La Palma, Spain. pp. 227-235.

[26] Tağluk ME, Ertuğrul ÖF. A joint generalized exemplar method for classification of massive datasets. Appl Soft Comput 2015; 36: 487-498.

[27] Duin RPW, Juszczak P, Paclik P, Pekalska E, de Ridder D, Tax DMJ. PR-Tools 4.0, A MATLAB Toolbox for Pattern Recognition. Technical report. Delft, the Netherlands: ICT Group, 2004.

[28] Smith JW, Everhart JE, Dickson WC, Knowler WC, Johannes RS. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: Annual Symposium on Computer Application in Medical Care; 6–9 November 1988; Washington, DC, USA. pp. 261-265.

[29] Bache K, Lichman M. UCI Machine Learning Repository. Irvine, CA, USA: University of California School of Information and Computer Science, 2013.

[30] Diaconis P, Efron B. Computer-intensive methods in statistics. Sci Am 1983; 248: 116-126.

[31] Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing 2006; 70: 489-501.

[32] Dorvlo AS, Jervase JA, Al-Lawati A. Solar radiation estimation using artificial neural networks. Appl Energ 2002; 71: 307-319.

[33] Berument H, Kiymaz H. The day of the week effect on stock market volatility. J Econ Finance 2001; 25: 181-193.

[34] Zhang SL. A study on the weekday effect and leverage effect on CSI-300 Index Futures Volatility-according to expanded conditional autoregressive range model of application. In: IEEE 2012 International Conference on Management Science and Engineering; 20–22 September 2012; Dallas, TX, USA. New York, NY, USA: IEEE. pp. 1522-1527.

[35] Chelton DB, Davis RE. Monthly mean sea-level variability along the west coast of North America. J Phys Oceanogr 1982; 12: 757-784.

[36] Watt JH, Van den Berg SA. Research Methods for Communication Science. New York, NY, USA: Allyn & Bacon, 1995.

[37] Schunk DH. Learning Theories: An Educational Perspective. 6th ed. Hoboken, NJ, USA: Pearson, 2012.

[38] Osherson DN, Smith EE. On the adequacy of prototype theory as a theory of concepts. Cognition 1981; 9: 35-58.

[39] Smith EE, Osherson DN. Conceptual combination with prototype concepts. Cognitive Sci 1984; 8: 337-361.

[40] Bouton ME, Moody EW. Memory processes in classical conditioning. Neurosci Biobehav R 2004; 28: 663-674.

[41] Sugar CA, James GM. Finding the number of clusters in a dataset. J Am Stat Assoc 2003; 98: 1-24.

[42] Pham DT, Dimov SS, Nguyen CD. Selection of K in K-means clustering. Proc IME C J Mech Eng Sci 219: 103-119.

[43] Perlich C, Swirszcz G. On cross-validation and stacking: building seemingly predictive models on random data. SIGKDD Explor 2011; 12: 11-15.

[44] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Stat Surv 2010; 4: 40-79.

[45] Qi M, Zhang GP. Trend time–series modeling and forecasting with neural networks. IEEE T Neural Networ 2008; 19: 808-816.

[46] Beck N, Katz JN. Random coefficient models for time-series–cross-section data: Monte Carlo experiments. Polit Anal 2007; 15: 182-195.

[47] Xu QS, Liang YZ. Monte Carlo cross validation. Chemometr Intell Lab Syst 2001; 56: 1-11.

[48] Arthur D, Vassilvitskii S. How slow is the k-means method? In: The Twenty-Second Annual Symposium on Computational Geometry; 5–7 June 2006; Sedona, AZ, USA. pp. 144-153.

[49] Mardia KV, Kent JT, Bibby JM. Multivariate Analysis. London, UK: Academic Press, 1979.

[50] Ratsch G, Onoda T, Muller KR. An improvement of AdaBoost to avoid overfitting. In: The Fifth International Conference on Neural Information Processing; 21–23 October 1998; Kitakyushu, Japan. pp. 506-509.

[51] Raymer ML, Doom TE, Kuhn LA, Punch WF. Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm. IEEE T Syst Man Cy B 2003; 33: 802-814.

[52] Ertuğrul ÖF, Tağluk ME. A novel machine learning method based on generalized behavioral learning theory. Neural Comput Appl (in press).

[53] Wang S, Li Z, Liu C, Zhang X, Zhang H. Training data reduction to speed up SVM training. Appl Intell 2014; 41: 405-420.

[54] Jensen R, Shen Q. Are more features better? A response to attributes reduction using fuzzy rough sets. IEEE T Fuzzy Syst 2009; 17: 1456-1458.

[55] Kumar CA, Srinivas S. Concept lattice reduction using fuzzy K-means clustering. Expert Syst Appl 2010; 37: 2696-2704.

[56] Dietterich, TG. Machine learning for sequential data: a review. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR); 6–9 August 2002; Windsor, ON, Canada. pp. 15-30.