Payame Noor University

Control and Optimization in Applied Mathematics (COAM)
Vol. 8, No. 1, Winter-Spring 2023 (33-53), ©2016 Payame Noor University, Iran

DOI. 10.30473/COAM.2023.66162.1221 (Cited this article)

Research Article

Effective Data Reduction for Time-Aware Recommender Systems

Hadis Ahmadian Yazdi[1], **Seyyed Javad Seyyed Mahdavi Chabok**[2,*], **Maryam Kheirabadi**[1]

[1]Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran.
[2]Department of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.

Received: November 21, 2022; **Accepted:** February 24, 2023.

Abstract. In recent decades, the amount and variety of data have grown rapidly. As a result, data storage, compression, and analysis have become critical subjects in data mining and machine learning. It is essential to achieve accurate compression without losing important data in the process. Therefore, this work proposes an effective data compression method for recommender systems based on the attention mechanism. The proposed method performs data compression on two levels: features and records. It is time-aware and based on time windows, taking into account users' activity and preventing the loss of important data. The resulting technique can be efficiently utilized for deep networks, where the amount of data is a significant challenge. Experimental results demonstrate that this technique not only reduces the amount of data and processing time but also achieves acceptable accuracy.

Keywords. Aggregate, Recommender systems, Feature selection, Correlation matrix, Dataset compression.

MSC. 90C34; 90C40.

* Corresponding author
h.ahmadian@um.ac.ir, mahdavi@mshdiau.ac.ir, m.kheirabadi@iau-neyshabur.ac.ir

1 Introduction

Over the last two decades, data has increased dramatically in many scopes such as in business, science, and social media. Therefore, the storage and transmission of data are growing at an enormous rate [14]. In other words, with the rapid development of the Internet and communication channels, the information explosion has become a serious problem [12]. A large amount of data and its dimensionality has become a severe task for machine learning and data mining [1, 6, 19]. The dimension of data has direct effects on the performance of the process, clustering, classification, regression, and time-series prediction. Moreover, a huge amount of data increases the cost of computation and storage pressure. In addition, there are valuable hidden data in the dataset that plays a crucial role in analyzing tasks. Normally, the original data contains irrelevant information [22]; which might cause misleading results and reduce accuracy. For example, in the K-nearest neighbor, the irrelevant features increase the distances between samples from the same class, which makes it more challenging to truly classify data [5, 13].

To prevent losing useful information and improve the performance of learning systems, efficient mechanisms should be considered to keep useful and important data [1, 4, 9]. Therefore, to solve ith the mentioned problem, an effective data compression algorithm for compressing and fetching useful data is proposed in this work. Generally, there are two types of compression, including lossy and lossless. In lossy mode, the initial data are different therefore it might lose importation data. On the other hand, the original data are the same as the retrieved in lossless compression. Although lossless compression reduces data without sacrificing precision, it has a limited ability to reduce scientific floating-point data because of the high entropy. Researchers employ lossy compression to reduce problems with the high entropy mantissa bits. By adding controlled error into the data, lossy compression can produce larger compression ratios. Many times, data has interesting subsets that researchers want to preserve with greater fidelity. Another technique for reducing data is data sampling, which saves a sparse subset of the data and discards the remainder to achieve higher levels of reduction [9]. In addition, compression can be classified into two categories of rows and columns [7]. In previous works, researchers concentrate more on the reduction of features. Which limited them to selecting a subset of data and discarding other records. However, in the proposed method, first, the data is reduced at the rows level by keeping the valuable data by introducing a "time-window" and using a correlation matrix for selecting relevant features. The feature selection techniques can decrease the amount of data and also speed up the analysis process. In other words, it is used to simplify the training phase and improve the quality of feature sets. Overall, it can significantly shorten the running time and improve accuracy [4, 5, 11].

It will be introduced in detail in the proposed method in part 2

1.1 Literature Review

As mentioned earlier, compression techniques can be classified into two categories: feature (column) or record (row) reduction. In previous works, most of the research focused on feature selection to reduce data volume and improve the performance of learning and data mining algorithms. Generally, to reduce the number of records, according to the problem raised in the research selected a subset of the main data is randomly or limited to a period of time or limited to a specific place. The rest of the data is discarded. In the proposed method, other dimensions such as records are also considered for this work.

Feature selection includes two main objectives, including the number of features and classification accuracy [22]. On the other hand, record reduction plays a significant role in reducing the spatial and temporal order of the mentioned algorithm. In this section, a brief overview of several features- and record-based selection schemes for data reduction proposed in recent years is discussed. In addition, the advantages and disadvantages of the techniques are presented.

In [9], a novel scheme was proposed for database compression based on the data mining technique. The redundant data which exist in the transaction database were eliminated and replaced with the mean of compression rules. Totally, it illustrated how association rules could be fetched using mining; that method has the improvement of both compression ratio and running time.

In [15], a threshold-based scheme has been used for feature selection in high-dimensional data. They used feature selection techniques for ranking attributes to find the strength of the relationship between attributes and classes. For this aim, each attribute was normalized and paired individually with the class. The method was compared by Naive Bayes and Support Vector Machine algorithms to learn from the training datasets. The results demonstrated the superiority of the technique compared to traditional standard filter-based feature selection.

In [17], the authors proposed a feature selection method based on Ant Colony Optimization (ACO) and Genetic Algorithm (GA). The method includes two main models visibility density and pheromone density models. In this approach, each feature is modeled as a binary bit, and each bit has two orientations for selecting and deselecting. In the experimental result, the scheme was compared with other evolutionary algorithms. The results prove the performance of the optimization algorithm for solving the feature selection problems.

In [5], a novel feature selection scheme was proposed based on genetic programming. In detail, a permutation strategy was employed to select features for high-dimensional symbolic regression. The regression result proved the method's efficiency compared to other traditional schemes by considering truly relevant features.

In [22], a novel unsupervised feature selection method was proposed based on Particle Swarm Optimization (PSO). In that way, two filter-based strategies were presented to speed up the convergence of the algorithm. The first filter was based on average mutual information, and the second one is based on feature redundancy. The first is employed to remove irrelevant and weakly relevant features; The second is applied to

improve the exploitation capability of the swarm. The experimental results illustrated the effectiveness of classification accuracy by reducing the number of features.

In [7], for the first time, the ensemble feature selection is modeled as a Multi-Criteria Decision-Making (MCDM) process. Used the VIKOR method to rank the features based on the evaluation of several feature selection methods as different decision-making criteria. Their proposed method first obtains a decision matrix using the ranks of every feature according to various rankers. The VIKOR approach is then used to assign a score to each feature based on the decision matrix. Finally, a rank vector for the features generates an output in which the user can select a desired number of features.

In [20] researchers trying to on modeling and predict flight delays. They propose a model for predicting flight delays based on Deep Learning (DL). They apply the proposed model undersampling method used on the U.S. flight dataset.

In [18] have designed a hybrid machine learning model to predict the short-term estimated arrival time in the terminal maneuvering area. The available dataset consists of July 2017 ADS-B records over the Beijing Capital International Airport (BCIA) TMA. By limiting the time and place, they have chosen that the studies are relatively small.

To date, various methods and data mining algorithms have been used to solve the issues of air traffic management and delay the minimization problems. In the paper [21], to increase the air traffic management accuracy and legitimacy used the design of the structure of a deep learning network. The Kaggle data include Summary information on the number of on-time, delayed, canceled, and diverted flights published in DOT's monthly Air Travel Consumer Report. The dataset that was recorded in 2015 has more than 5,800,000 flights. These flights are described according to 31 variables. Due to a large amount of data, this paper has used a sub-dataset of Kaggle data including 1,00,000 records and 15 features to validate the proposed method.

In [16] discussed an iterative machine and deep learning approach for aviation delay prediction. The dataset required for training and testing is obtained from Kaggle.com. This dataset contains 30 features. Each of these attributes is of great importance. There are 10,000 samples in the dataset. Due to the limitation of the processing power of the system used to perform the experiment, the concept of stratified random sampling has been used to select the training data set. The entire dataset is divided into layers and from each layer, training samples are randomly selected. The main advantage of using stratified sampling is that it represents the entire population and reduces sample selection bias. Different layers represent different classes or categories in the data. For this purpose, the concept of stratified sampling should be reduced to two thousand samples.

1.2 Key Contributions

In recommendation systems, the activity of users, and the related data are saved and stored over a period of time. In this proposed method, the time of user activity is divided into different scopes of time which we call "Time windows", which can be weekly, monthly, or quarterly, from the beginning of the activity when the activity

is recent the value of the time windows increases and when it goes further the value decrease also the duration of these windows are vital too. If the duration gets too big it could have a negative effect on the result, which will be more explained in the result part. In the process of compression part of the data was last or ignored, in this method, instead of losing or ignoring part of the data, the data will get weight and value. According to time windows, which overall increase the accuracy of the compression.

The rest of this paper is organized as follows. The details of the proposed method are described in Section 2. The experimental results, analysis, and performance comparison are given in Section 3. Finally, the conclusions and future scope are shown in Section 4.

2 Proposed Method

As discussed earlier, the issue of previous works in compression was randomly or selectively losing and deleting part of data which could be critical in decision-making. In this proposed method no data will be lost or deleted instead, weighted time windows have been introduced (Figure 1) by reviewing the user activity. Also, the compression operation is employed in both record and feature levels. The activity of the user can be reading a web page, downloading PDFs, watching educational movies, etc. Besides, for improving the performance of recommender schemes, the value of users' behavior is effective in the process. In this way, current activities are more substantial in comparison to the previous.

2.1 Data preprocessing

The steps in the preprocessing can be seen in Figure 2. First, we extract the provided resources, student features, courses held, and student performance and evaluation in each course from the OULAD standard database. After merging the data, we proceed to categorize and map the features by Converting the string values of the quantitative variables in the database to numeric values, deleting the empty or incorrect data, and normalizing the features.

By (1), we proceed to normalize the feature in the domain $[0,1]$.

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

$x_{\min}$ represents the minimum value, $x_{\max}$ is the maximum value, x^* is the normalized value, and x is the original data.

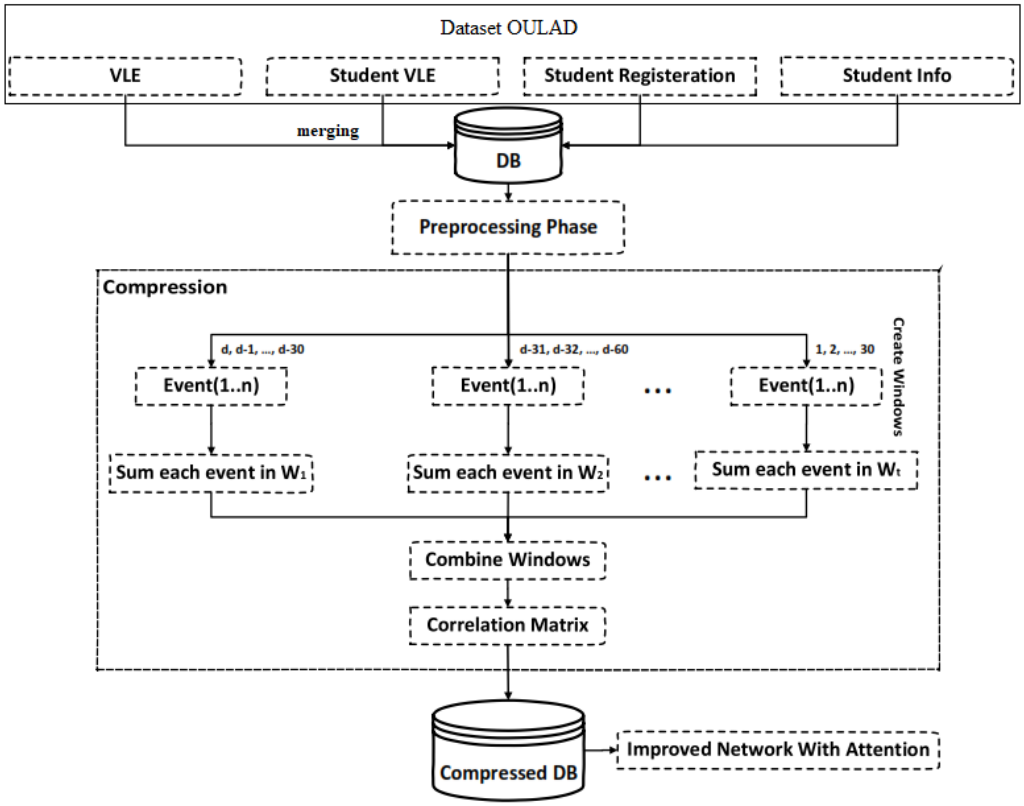

Figure 1: The block diagram of the proposed method.

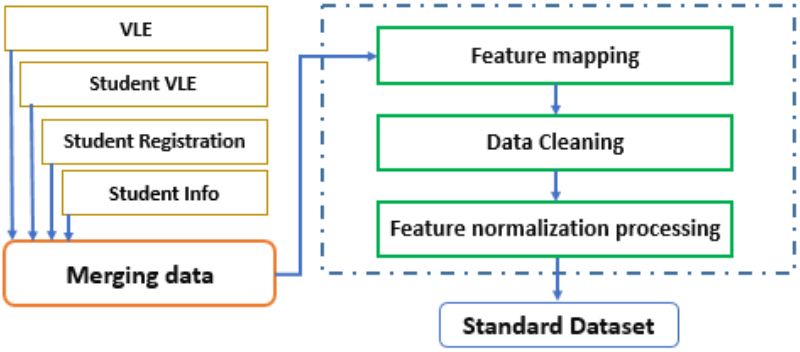

Figure 2: Data Preprocessing Steps.

2.2 Database

In the presented work, the Open University Learning Analytics Dataset (OULAD) is used for analyzing algorithms [8, 23]. This dataset contains demographical data, including students' information, their attended courses, and the final results of each course. In

https://analyse.kmi.open.ac.uk/open_dataset

detail, it contains the students' interactions with Virtual Learning Environment (VLE) for seven selected courses. The dataset includes 22 modules of over 30,000 students. The data is fetched by the daily summaries of student clicks on several resources. In OULAD, the tables are connected using unique identifiers; It should be noted, that the tables are stored in the CSV format. The utilized files are briefly described below:

- **Assessments:** These contain information about assessments in module presentations.
- **StudentInfo:** This file holds demographic information about the students together with their results.
- **StudentVle:** The file includes information about each student's interactions with the elements in the VLE.
- **StudentAssessment:** This file contains the results of students' assessments.

Table 1, shows an example of the values available in the original database (OULAD) [8, 23] that are mapped to numerical values in Table 2 (designed by the researcher)

Table 1: A sample of database data before mapping[8, 23]

code_module	code_presentation	id_student	gender	highest_education	age_band	final_result	score_mean	id_site	date	sum_click
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546669	-5	16
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546662	-5	44
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546652	-5	1
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546668	-5	2
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546652	-5	1
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546670	-7	2
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546671	-7	2
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546669	-5	16
AAA	2013J	11391	M	HE Qualification	55<=	Pass	82	546662	-5	44

Table 2: The values of features mapped to the number

code_module		code_presentation		age_band		gender		highest_education		final_result	
AAA	0.1	2013B	540	0-35	0.1	F	0.1	A Level or Equivalent	0.1	Distinction	0.1
BBB	0.2	2013J	720	35-55	0.2	M	0.2	HE Qualification	0.2	Fail	0.2
CCC	0.3	2014B	180	55<=	0.3	-	-	Lower Than A Level	0.3	Pass	0.3
DDD	0.4	2014J	360	-	-	-	-	No Formal equals	0.4	Withdrawn	0.4
EEE	0.5	-	-	-	-	-	-	Post Graduate Qualification	0.5	-	-
FFF	0.6	-	-	-	-	-	-	-	-	-	-
GGG	0.7	-	-	-	-	-	-	-	-	-	-

2.3 Symbols

In the recommendation system, one aspect of the choice is the user and the other vital aspect is the selected option, which could be a course by a student, a movie by a viewer, a track of music for a listener, or a meal by an eater and so on. Assuming U represents a collection of users and I is the items chosen by the user. In this scheme, the main

goal is extracting the interests and priorities of users by looking at User-Item interactive events. For example, clicking on an educational source is considered an action for users. Also, for each user, $u \in U$ is sequential time windows as $Wu = \{w_1u, w_2u, \ldots, w_tu\}$ which t represents the total number of time windows; Also, w_tu shows a collection of smaller time units as $w_tu = \{d_1, d_2, \ldots, d_x\}$ where x indicates the length of time windows. Moreover, the related items of the user (u) in time windows (t) express by w_tu. There are some events in each time window as $\{et, iu \in \ \mathrm{Rm} | i = 1, 2, \ldots, t\}$ that iu and et describe the event i in time units of windows time (d_x). Something else which should be mentioned is that user u interacts with the item $|i \in I|$ in each event.

$|u \in U|$
$Wu = \{w_1u, w_2u, \ldots, w_tu\}$
$w_tu = \{d_1, d_2, \ldots, d_x\}$
$\{et, iu \in \ \mathrm{Rm} | i = 1, 2, \ldots, t\}$
$|i \in I|$
$Rd_x = \sum_{k=0}^{N} i_k$
$\mathrm{Rw}_tu = \sum_{k=0}^{N} i_k$
$\mathrm{Length\_time\_window} = Td_x - Td_1$
$i' = \mathrm{Rw}_tu / \ \mathrm{Length\_time\_window}$

2.4 Compression in record level

First, one window (or limited number) is considered as a background. In this way, the user's different activities appearing in a time window are observed. As mentioned, the different weights based on the average of constituent days are assigned for each window. With the help of these techniques, each user's activity with attention to the category of the time window is marked by proportional weight. Hence, the smaller weights can be considered for distant time windows. On the contrary, due to the crucial role of near-time windows, the bigger weights are selected for them. In the following, the weight of the feature is multiplied by the summation of the corresponding activity; Then, the outcome is aggregated with the results of the rest of the windows. Finally, each feature that is considered as a background is repeated only once in the section; and in the new field, it maintains the frequency of the merged features.

2.5 Compression in feature level

After compression at the record level, the feature selection with the correlation matrix technique (see (2)) is applied to further compress data at the feature level. We use the Correlation Matrix with Pearson which measures linear dependence between two variables [10]. After implementing the Correlation Matrix with Pearson, we get a matrix $n \times n$, with n being the number of features. Matrix values are the Correlation Coefficient values ranging from -1.0 to $+1.0$ with -1.0 being a total negative correlation, 0.0 being no correlation and $+1.0$ being a total positive correlation. With the help of this strategy,

the feature pairs with the highest correlation value are selected. For more info about the correlation matrix refer to [18, 10].

$$r = \frac{\sum (x - m_x)\left(y - m_y\right)}{\sqrt{\sum (x - m_x)^2 \sum \left(y - m_y\right)^2}} \tag{2}$$

1. x and y are two vectors of length n,
2. m_x and m_y corresponds to the means of x and y, respectively.

In Table 3, you can see an example of the values after applying the compression technique to the values in the database.

The network's performance is significantly improved by decreasing the number of time windows and features. Notice that the Attention strategy is applicable in most engineering problems such as information retrieval, machine vision, recommender systems, etc. [**?**, 3].

Table 3: A sample of the final database after mapping and compression.

code_module	code_presentation	id_student	gender	highest_education	age_band	final_result	id_site	date	sum_click	max_mean_score
0.1	720	11391	0.2	0.2	0.3	0.3	82	715	16	82
0.1	720	11391	0.2	0.2	0.3	0.3	82	715	44	82
0.1	720	11391	0.2	0.2	0.3	0.3	82	715	1	82

The details of the algorithm are summarized in the following steps:

1. First, the required preprocessing, including data cleaning, removal of missing data, etc., is done (Algorithm 3).
2. Determine the length of time windows (for instance, one month).
3. Do the following steps for each user.
   (a) Do the following steps for each time window.
      i. Compute the average length of the time window.
      ii. Do the following steps for each item.
      iii. Compute the whole activities (clicking) of the special item for determining time windows.
      iv. Now, the results are divided by the average of step 5. With this technique, the near and distant time windows are categorized based on considering different weights.
      v. In this step, the new candidate resource which appears in current time windows is appended to the collection of previous time windows. In the same candidate case, the computed value of the current windows is added to the previous value of resources.
      vi. Create a new field to maintain the number of integrated features merged in each window.

4. After compression at the record level, the feature selection with the correlation matrix technique is applied to further compress data at the feature level. This technique contains a table that demonstrates correlation coefficients between the collection of features. With the help of this strategy, the feature pairs with the highest correlation value are selected. For more info about the correlation matrix refer to [14, 18].

5. Apply the Correlation Matrix Algorithm technique to the database. To do so, the compression step must be done in the columns to compress data in column diminutions based on the importance of the available features.

6. Finally, the results are considered as input, and the output of the system is illustrated as the recommended list. In other words, the order of recommended resources is sorted by the network.

Generally, unlike presented schemes in the last decades that concentrate on feature compression (feature selection), this work presents a compressing strategy in terms of record (row). With the help of these mechanisms, the time complexity and the training time are significantly improved. Meanwhile, the accuracy of the system is optimized compared to previous works. These claims are proved with the experimental results reported in the next section.

By reducing the number of records, the network speed is increased. Furthermore, the necessary hardware resources such as Ram and GPU to implement the deep learning network are reduced. The process of training a network with multimillion-record datasets on Kulb's with high-level hardware takes several days; Moreover, We are facing the challenge of memory shortage due to working on datasets with a tabular structure requiring all table values to be fetched in memory to perform the network training process. By using techniques such as data generators to manage the memory challenge, the time spent on training will be much more than before. Totally, it may not be possible to use techniques such as data generators in some types of networks, such as DBN, which encounter the enlargement of the weight matrix during the training process, and researchers will have to select several records.

3 Experimental Results

In the following, we will first check the effectiveness of our proposed compression on the amount of data volume reduction at two levels, attributes, and records. The number of records which is used in this work after the initial stages of pre-processing is 10403715 and each record consists of 12 fields. We have evaluated the data compression effect of our algorithm in several different time window sizes. It can be seen in the WIN Count column of Table 4, that the number of windows matched on data records with different window lengths. As expected, the larger window length leads to a smaller number of windows; Due to the Correlation Matrix algorithm and the dataset fields, the compression output has a constant value equal to 11 in the column dimension in the whole evaluation process. While the amount of data compression in the row level

Algorithm 3 The pseudo-code of preprocessing Data

Input: The CSV files of Student Info, Student VLE, Student Assessment, Assessments.
Output: Cleaned Data.

```
procedure Preprocessing Data
    Read input files
    Merge files
    Remove missing values
    Convert string values to number
    Update date from "far to now" to "now to far"
    Create compressed Dataset with following attributes
        [id student, age band, gender, highest education, number of previous attempts,
    final result, code module, date, date count, mean of sum click]
    return Optimized (Cleaned) Tables
end procedure
```

depends on the selected time window length; By selecting the larger time window, the more compression and the fewer records resulted from the compression operation. On the other hand, as can be seen in the "Time Data dimensions have reduced (sec)" column of Table 4, since this technique proposed is separately applied to each window, the larger window length leads to a smaller number of windows and finally less compression time.

Table 4: The effect of compression on the number of rows and columns

Original dataset	Number of rows and columns =10,543,682*12					
	win count	Time Data Label(sec)	Label Count	Time Data dimensions have reduced(sec)	Record Test	Record Train
	-	125.22121	562	0	3479416	7064266
Dataset Compress Win7	Number of rows and columns =5167599*11					
	win count	Time Data Label(sec)	Label Count	Time Data dimensions have reduced(sec)	Record Test	Record Train
	116	123.32143	252	48.17566	1705308	3462291
Dataset Compress Win 14	Number of rows and columns =4239764*11					
	win count	Time Data Label(sec)	Label Count	Time Data dimensions have reduced(sec)	Record Test	Record Train
	58	122.14785	453	23.96710	1399123	2840641
Dataset Compress Win 30	Number of rows and columns =3396653*11					
	win count	Time Data Label(sec)	Label Count	Time Data dimensions have reduced(sec)	Record Test	Record Train
	27	132.04799	1123	12.85002	1120896	2275757
Dataset Compress Win 60	Number of rows and columns = 2816927*11					
	win count	Time Data Label(sec)	Label Count	Time Data dimensions have reduced(sec)	Record Test	Record Train
	14	128.29239	3754	8.63251	929586	1887341

In the following, by applying the proposed technique with different window lengths and compressing the input data of several different deep learning networks, we have investigated the effect of the proposed technique on the Loss and Accuracy resulting from the training and testing of the deep learning network. In this research, we have trained and evaluated five architectures such as LSTM, GRU, LSTM + Attention, GRU + Attention, and Bi-LSTM in 3 different window lengths. As shown in Table 5, in the first step for the different architectures, we have trained and tested the original and uncompressed data in 50 epochs. In the next steps, during the 7, 14, and 30 day windows, we have trained and tested the same architectures with 100 epochs. In this way, plus using fewer hardware resources such as RAM, you will see that although the number of epochs has doubled, it has saved much time. Moreover, As you can see in Table 6, by comparing the results of the main data and the accuracy and loss

after applying data compression in the training and testing phase of the implemented models (Table 5), it can see that the network with the length of the window equal to 7 and 14 learns with high speed and acceptable accuracy. The data volume is almost halved during a window length equal to 7; despite halving the data and increasing the execution speed, The average accuracy of training and testing of the implemented models is maintained. Also, for a window with a length of 14, when our data volume has almost reached 40%, the accuracy has dropped by 0.10%. As you can see in Table 5 and Figure 5, better results have been obtained in the structures that use the attention layer in the network architecture. Therefore, it is highly recommended to use the attention layer in the network architecture. The attention mechanism has been used in many tasks such as information retrieval, machine translation, computer vision, and recommender. The main idea of these techniques is to learn accurate (normalized) weights to assign to a set of features. Thus, higher weights indicate that the relevant features contain more important information for the relevant task. This technique calculates the importance of each item to a particular user and feeds this information to the system to indicate the user's interests and preferences. Therefore, with the help of this strategy, important parts of the input that can be effective in the output result are automatically identified and given more weight. Ultimately, it helps in better results. We have trained our networks on two platforms including a PC with Nvidia GeForce GTX 1060 6GB and Colab. As you can see in the "Time (seconds)" column of Table 5, the training speed would have been much higher by running on the colab platform; Due to the limitations of this type of free platform, network training with uncompressed data was not provided and we faced a shortage of memory. As we have already mentioned, researchers sometimes miss the possibility of network training due to the high volume of data and the impossibility of accessing the appropriate hardware. Our proposed method is provided by choosing the appropriate window length. The different parts of Figures 3, 4, 5, 6, and 7 show the training and loss diagrams of the mentioned architectures examined in Table 5 . By looking in detail, it can be seen that after data compression, the model reaches high accuracy in lower periods than the original data. Over-fitting has occurred Only in the Bi-Lstm structure with window lengths of 7 and 14. The pulsation in the graph shows that the model is trying to improve itself. In other architectures, with all window lengths, overfitting did not happen. Also, the downward trend in the slope of the Loss charts at the early epoch indicates the appropriateness of the selected learning rate in the training process. By increasing the length of the time window when compressing the existing data. Part of the hidden data is lost, as a result, the accuracy of the model trained with this data decreases.

Table 5: Investigating the effect of selected window length in accuracy and loss of training and testing phases

		Epoch 50		Original dataset				Epoch 100		Compress Win 7			
Structure of Network		Time (seconds)		loss		accuracy		Time (seconds)		loss		accuracy	
		PC	Colab	Val	train	Val	train	Colab	PC	Val	train	Val	train
LSTM	train	115395	low Memory or Time	0.3231	0.3185	86.52%	86.45%	15853	65003	0.2731	0.2474	87.67%	88.72%
	test	1933		-		90%		686	983	-		88%	
GRU	train	154456		0.2742	0.2434	88.09%	89.3 %	42546	82781	0.2022	0.194	90.48%	90.87 %
	test	2486		-		89%		836	1558	-		91%	
LSTM+ Attentiom	train	172034		0.1998	0.172	91.22%	92.29%	15108	90539	0.2472	0.2302	88.55%	89.34%
	test	1970		-		94%		720	1056	-		89%	
GRU + Attentiom	train	205520		0.479	0.268	80.9%	81.5%	48011	93042	0.2207	0.2086	89.77%	90.34%
	test	2518		-		81%		1005	1756	-		90%	
Bi-LSTM	train	188240		0.4678	0. 3575	82.99%	86.09%	99400	160885	0.5672	0. 4872	81.04%	82.56%
	test	2020		-		86%		875	1152	-		82%	
		Epoch 100		Compress Win 14				Epoch 100		Compress Win 30			
Structure of Network		Time (seconds)		loss		accuracy		Time (seconds)		loss		accuracy	
		PC	Colab	Val	train	Val	train	Colab	PC	Val	train	Val	train
LSTM	train	53769	12529	0.647	0.5591	75.06%	77.35%	10418	43670	1.2694	1.1116	55.92%	59.86%
	test	762	587	-		77%		296	556	-		60%	
GRU	train	68944	30388	0.3681	0.3352	83.7%	85.08%	25549	55505	0.6365	0.7173	70.62%	73.51%
	test	1347	583	-		85%		311	1153	-		72%	
LSTM+ Attentiom	train	86345	11982	0.5188	0.4741	78.64%	80.29%	9816	59918	0.8063	0.7019	68.11%	72.19%
	test	839	654	-		80%		412	702	-		72%	
GRU + Attentiom	train	79475	42066	0.515	0.4932	79.11%	80.01%	35313	60301	0.6546	0.5869	72.83%	75.3%
	test	1520	875	-		80%		401	1293	-		75%	
Bi-LSTM	train	132128	59400	1.155	1.0703	67.23%	68.6%	50733	106813	1.1911	0.9882	57.21%	62.18%
	test	912	756	-		69%		590	807	-		62%	

Table 6: Summarizing the impact of compression in terms of accuracy, saving memory and time. (WI means the length of window and DC means the length of Dataset Compress)

Dataset	Average accuracy in trained models	Average execution time in PC for trained architectures (seconds)	Average execution time in Colab for trained models (seconds)	Percentage reduction of data records after compression
Original	84657.2	NAN	0	0.88
Compression (WI=7)	49875.5	22504	50.98867%	0.88
Compression (WI=14)	42604.1	15982	59.78858%	0.78
Compression (WI=30)	33071.8	13383.9	67.78494%	0.682

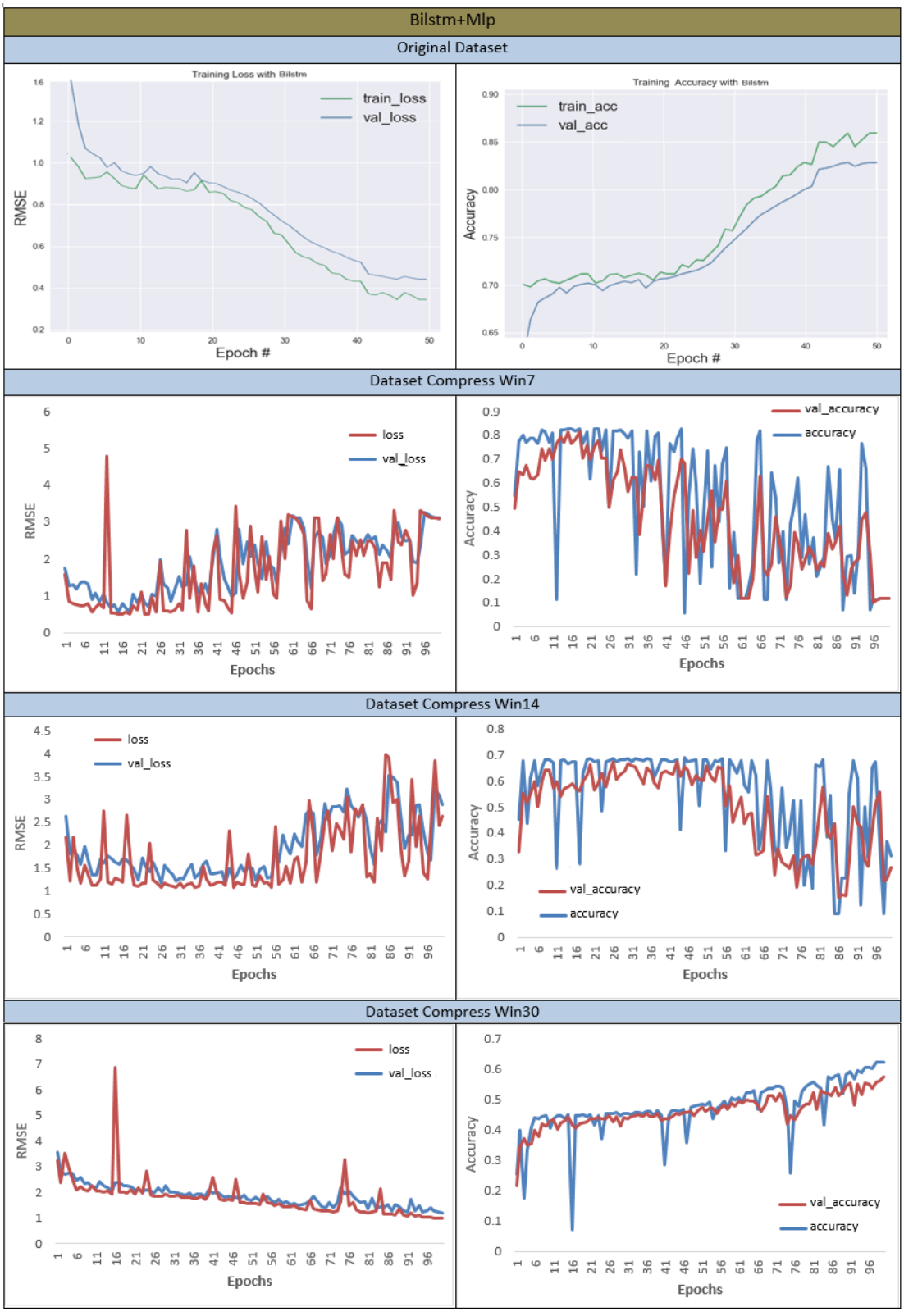

Figure 3: Bi-LSTM.

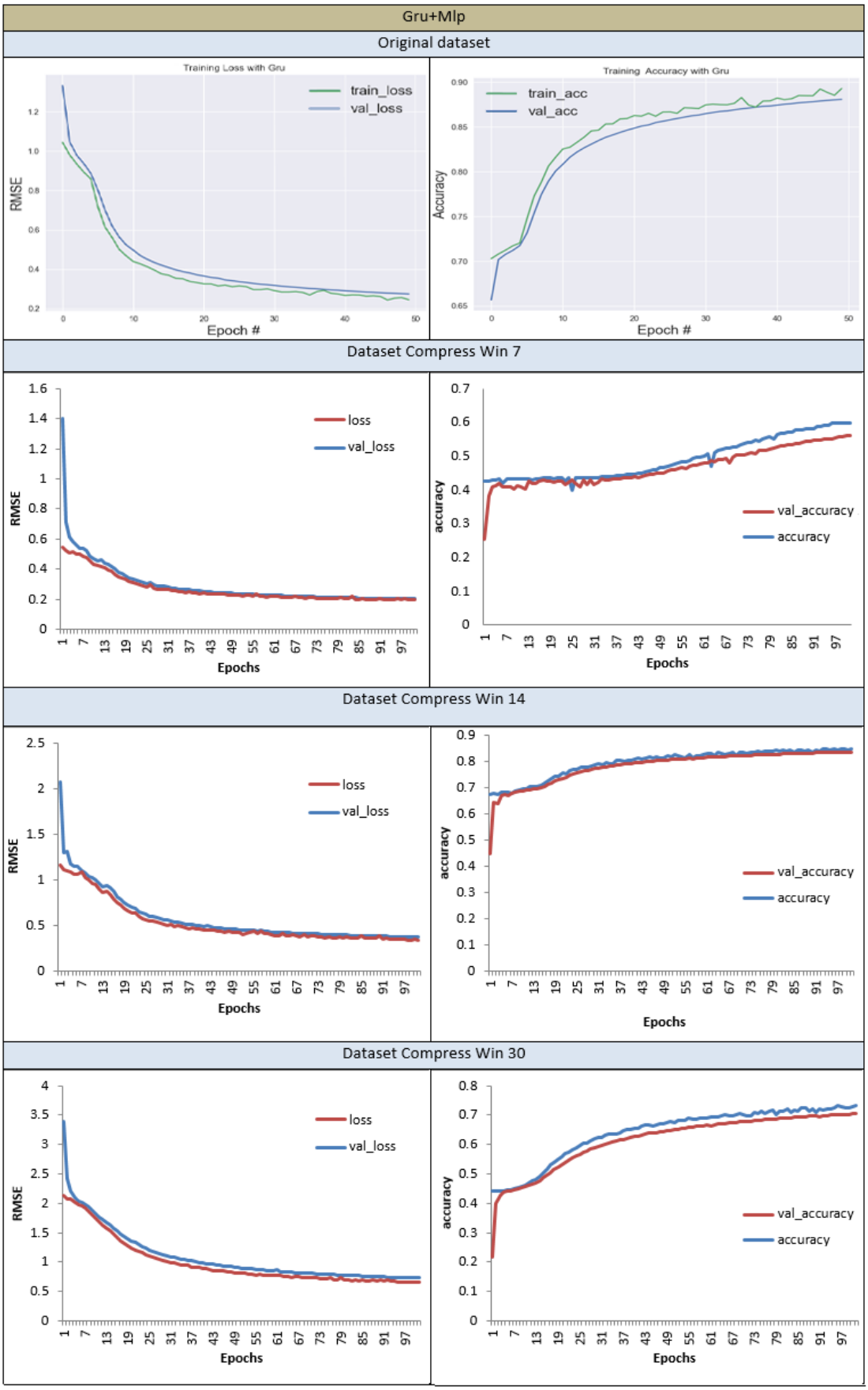
Gru+Mlp
Original dataset
Training Loss with Gru
train_loss
val_loss
RMSE
Epoch #
Training Accuracy with Gru
train_acc
val_acc
Accuracy
Epoch #
Dataset Compress Win 7
loss
val_loss
RMSE
Epochs
val_accuracy
accuracy
accuracy
Epochs
Dataset Compress Win 14
loss
val_loss
RMSE
Epochs
val_accuracy
accuracy
accuracy
Epochs
Dataset Compress Win 30
loss
val_loss
RMSE
Epochs
val_accuracy
accuracy
accuracy
Epochs

Figure 4: GRU.

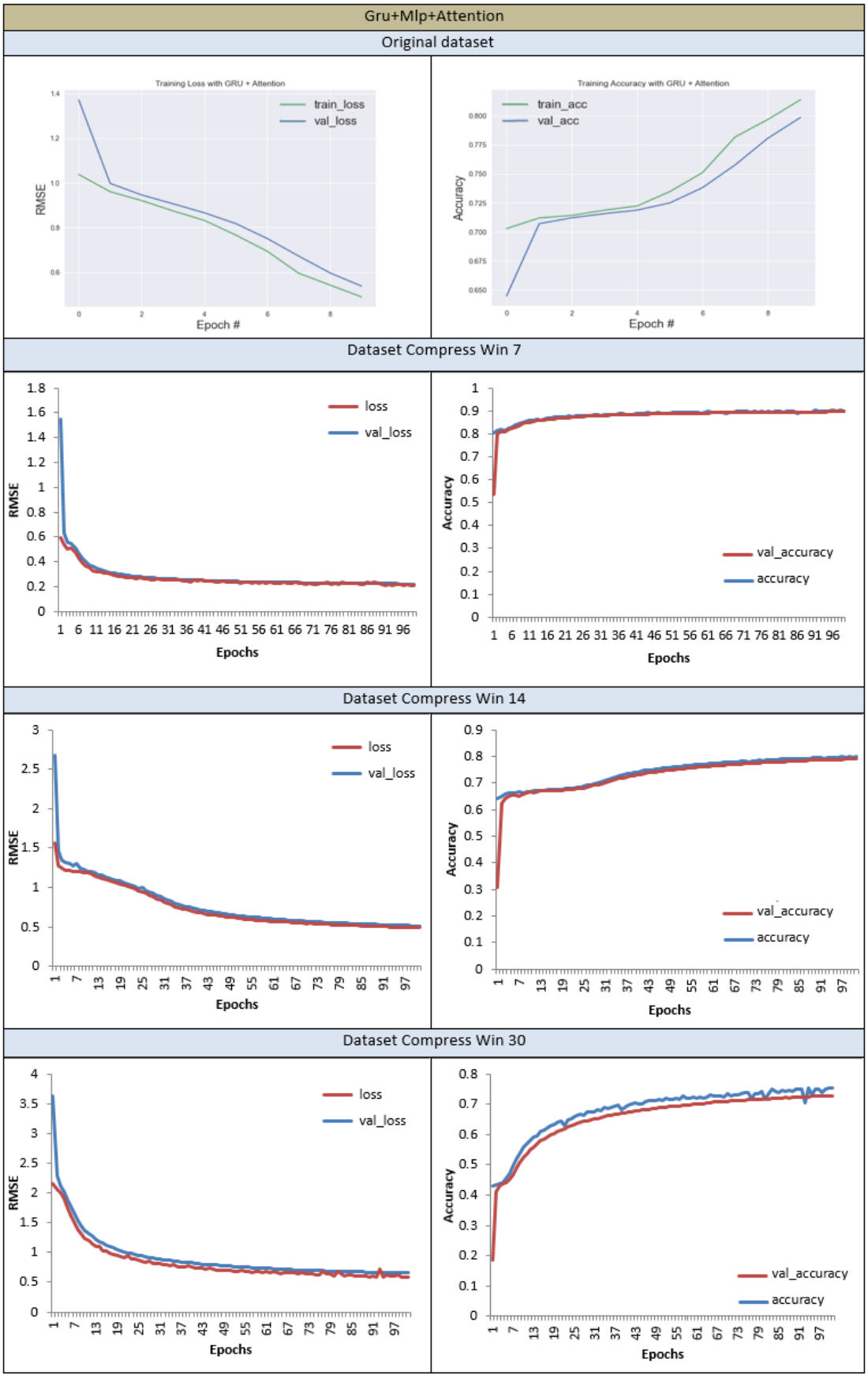

Figure 5: GRU+Attention.

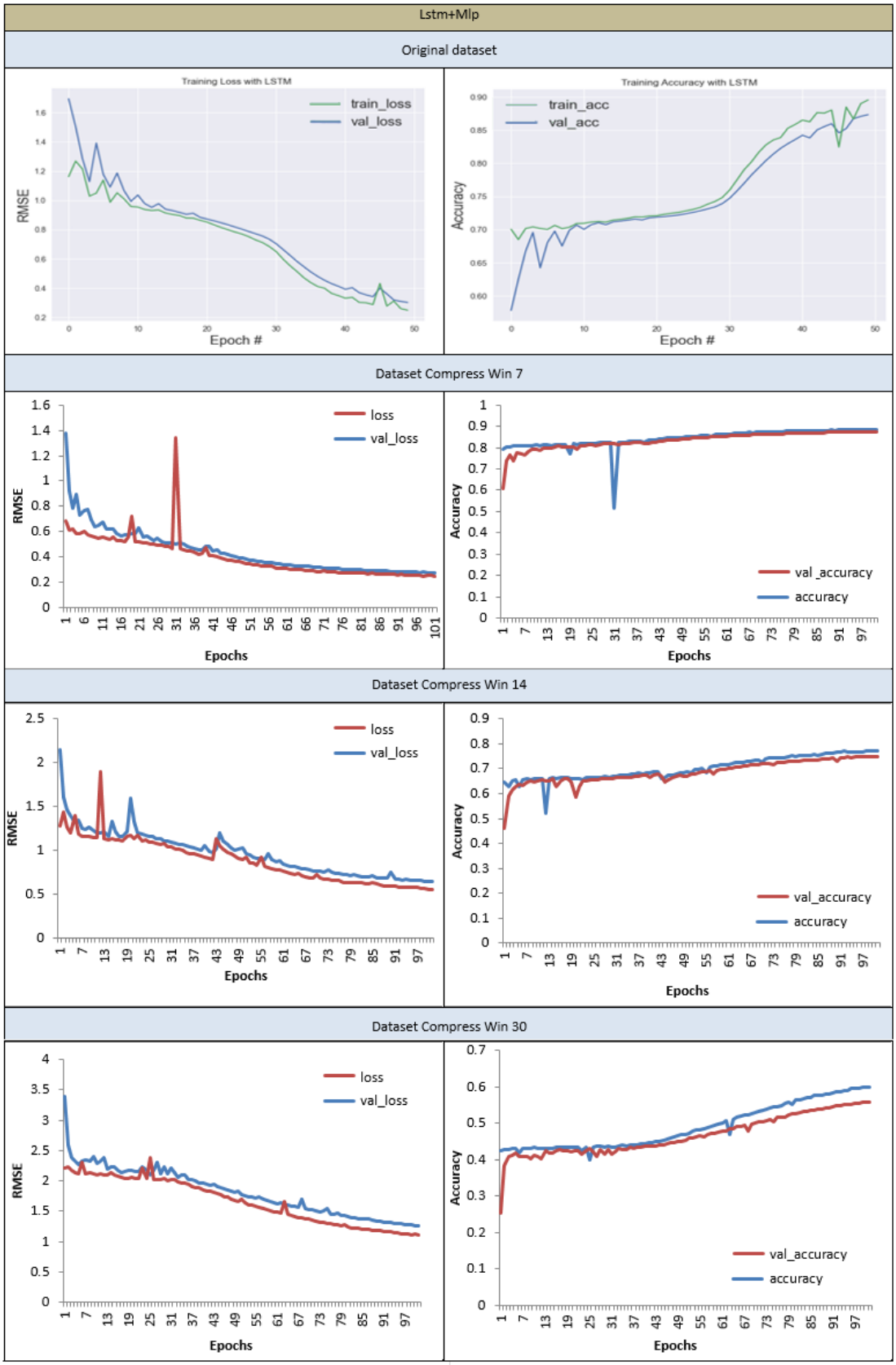

Figure 6: LSTM.

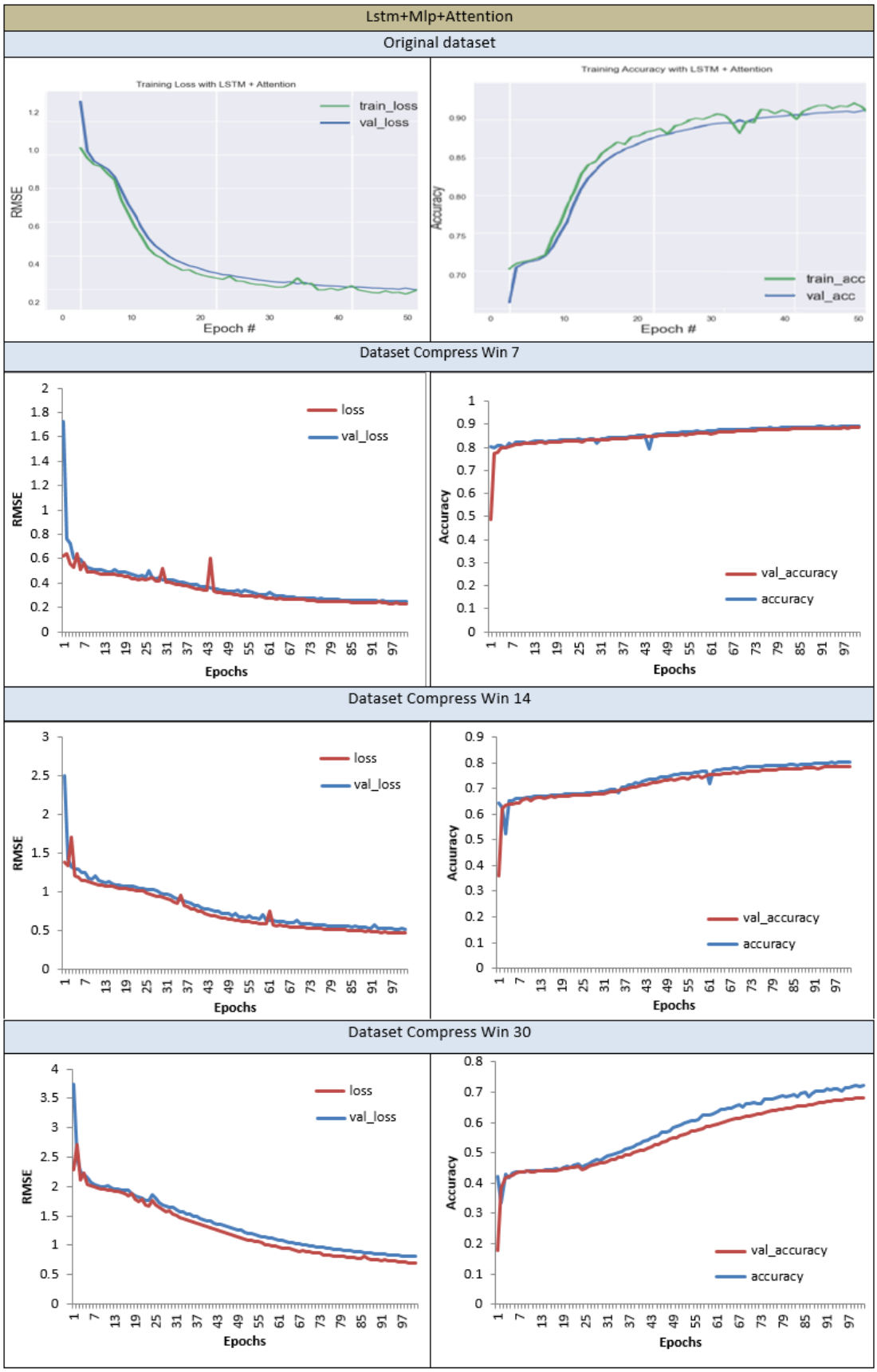

Figure 7: LSTM+Attention.

4 Conclusion and Future Works

The exponential growth of data has led to an increased interest in compression schemes that can effectively reduce storage and processing requirements. At the same time, neural networks have become a popular tool for solving various problems in different domains. However, the large amount of data required for training these networks presents a significant challenge. To address this issue, we propose a novel compression method for tabular datasets that can be used in conjunction with deep learning models. Unlike previous approaches that only compressed data at the feature level, our method performs compression at both the row and column levels, resulting in more efficient use of space. Additionally, our approach is time-aware and takes into account users' activity by using time windows, ensuring that important data is not lost during the compression process. One of the key advantages of our method is that the compressed data can be used directly for training without requiring additional steps. Experimental results demonstrate that our approach achieves high accuracy while significantly reducing the processing time and memory requirements. In future work, we plan to explore feature augmentation and assembly techniques to further improve the compression process.

Declarations

Availability of supporting data

All data generated or analyzed during this study are included in this published paper.

Funding

This study received no funds, grants, or other financial support.

Competing interests

The authors declare no competing interests that are relevant to the content of this paper.

Authors' contributions

The main manuscript text is collectively written by all authors.

References

[1] Ahmadian Yazdi H., Seyyed Mahdavi Chabok S.J., Kheirabadi M. (2021). "Dynamic educational recommender system based on improved recurrent neural networks using attention technique", Applied Artificial Intelligence, 1-24.

[2] Bonyani M. Ghanbari M., Rad A. (2022). "Different gaze direction (DGNet) collaborative

learning for iris segmentation", Available at SSRN 4237124.

[3] Bonyani M., Soleymani M. (2022). "Towards improving workers' safety and progress monitoring of construction sites through construction site understanding", arXivpreprintarXiv: 2210.15760.

[4] Brezocnik L., Fister I., Podgorelec V. (2018). "Swarm intelligence algorithms for feature selection: A review", Applied Sciences, 8 (9), 1521.

[5] Chahboun S., Maaroufi M. (2022). "Performance comparison of K-nearest neighbor, random forest, and multiple linear regression to predict photovoltaic panels' power output", Advances on Smart and Soft Computing, Springer, 301-311.

[6] Chen Q., Zhang M., Xue B. (2017). "Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression", IEEE Transactions on Evolutionary Computation, 21(5), 792-806.

[7] Hashemi A., Dowlatshahi M. B., Nezamabadi-pour H. (2022). "Ensemble of feature selection algorithms: a multi-criteria decision-making approach", International Journal of Machine Learning and Cybernetics, 13 (1), 49-69.

[8] Kuzilek J., Hlosta M., Zdrahal Z. (2017). "Open university learning analytics dataset", Scientific data, 4, 170171.

[9] Lee C.-F., Changchien S. W., Wang J.-J. (2006). "A data mining approach to database compression", Information Systems Frontiers, 8 (3), 147-161.

[10] Luong H. H., Tran T. T., Van Nguyen N., Le A. D., Nguyen H. H. T., Nguyen K. D., Tran N. C., Nguyen H. T. (2022). "Feature selection using correlation matrix on metagenomic data with pearson enhancing inflammatory bowel disease prediction", in: International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17-18 December, Universiti Teknologi Petronas, Malaysia, Springer, 1073-1084.

[11] Nguyen B. H., Xue B., Zhang M. (2020). "A survey on swarm intelligence approaches to feature selection in data mining", Swarm and Evolutionary Computation, 54, 100663.

[12] Pan Y., He F., Yu H. (2019). "A novel enhanced collaborative autoencoder with knowledge distillation for top-N recommender systems", Neurocomputing, 332, 137-148.

[13] Prasetyawan D., Gatra R. (2022). "Algoritma K-nearest neighbor untuk memprediksi prestasi mahasiswa berdasarkan latar Belakang pendidikan dan ekonomi", Jurnal Informatika Sunan Kalijaga, 7 (1), 56-67.

[14] Uthayakumar J., Vengattaraman T., Dhavachelvan P. (2018). "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications", Journal of King Saud University-Computer and Information Sciences.

[15] Van Hulse J., Khoshgoftaar T. M., Napolitano A., Wald R. (2012). "Threshold-based feature selection techniques for high-dimensional bioinformatics data", Network modeling analysis in health informatics and bioinformatics, 1 (1-2), 47-61.

[16] Venkatesh V., Arya A., Agarwal P., Lakshmi S., Balana S. (). "Iterative machine and deep learning approach for aviation delay prediction", in: 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), IEEE, 562-567.

[17] Wan Y., Wang M., Ye Z., Lai X. (2016). "A feature selection method based on modified binary coded ant colony optimization algorithm", Applied Soft Computing, 49, 248-258.

[18] Wang Z., Liang M., Delahaye D. (2018). "A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area", Transportation Research Part C: Emerging Technologies, 95, 280-294.

[19] Xue B., Zhang M., Browne W. N., Yao X. (2015). "A survey on evolutionary computation approaches to feature selection", IEEE Transactions on Evolutionary Computation, 20(4), 606-626.

[20] Yazdi M. F., Kamel S. R., Chabok S. J. M., Kheirabadi M. (2020). "Flight delay prediction based on deep learning and Levenberg-Marquart algorithm", Journal of Big Data, 7, 1-28.

[21] Yousefzadeh Aghdam M., Kamel Tabbakh S. R., Mahdavi Chabok S. J., Kheyrabadi M. (2021). "Optimization of air traffic management efficiency based on deep learning enriched by the long short-term memory (LSTM) and extreme learning machine (ELM)", Journal of Big Data, 8 (1), 1-26.

[22] Zhang Y., Li H.-G., Wang Q., Peng C. (2019). "A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection", Applied Intelligence, 49 (8), 2889-2898.

[23] "Oulad: Open university learning analytics dataset, https://www.kaggle.com/vjcalling/oulad-open-university-learning-analytics-dataset.

How to Cite this Article:
Ahmadian Yazdi, H. Seyyed Mahdavi Chabok, S.J., KheirAbadi, M. (2023). "Effective data reduction for time-aware recommender systems", Control and Optimization in Applied Mathematics, 8(1): 33-53. doi: 10.30473/COAM.2023.66162.1221