Converting Time Series Data to Numeric Representations Using Alphabetic Mapping and k-mer strategy

ABSTRACT

In the realm of data analysis and bioinformatics, representing time series data in a manner akin to biological sequences offers a novel approach to leverage sequence analysis techniques. Transforming time series signals into molecular sequence-type representations allows us to enhance pattern recognition by applying sophisticated sequence analysis techniques (e.g. k-mers based representation) developed in bioinformatics, uncovering hidden patterns and relationships in complex, non-linear time series data. This paper proposes a method to transform time series signals into biological/molecular sequence-type representations using a unique alphabetic mapping technique. By generating 26 ranges corresponding to the 26 letters of the English alphabet, each value within the time series is mapped to a specific character based on its range. This conversion facilitates the application of sequence analysis algorithms, typically used in bioinformatics, to analyze time series data. We demonstrate the effectiveness of this approach by converting real-world time series signals into character sequences and performing sequence classification. The resulting sequences can be utilized for various sequence-based analysis techniques, offering a new perspective on time series data representation and analysis.

KEYWORDS

Time Series Data, Sequence Analysis, Alphabetic Mapping, Bioinformatics, Data Transformation, Protein Sequence Representation, Signal Processing, Data Encoding, Character Sequences, Time Series Classification

ACM Reference Format:

. 2018. Converting Time Series Data to Numeric Representations Using Alphabetic Mapping and k-mer strategy. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Time series data is ubiquitous in various fields such as finance [15], healthcare [19], environmental monitoring [1, 22], and industrial processes [16]. Traditionally, analyzing time series data involves techniques like statistical modeling, machine learning, and signal processing. However, these methods often struggle with capturing complex, non-linear patterns inherent in many time series datasets [20]. In bioinformatics, sequence analysis has proven to be highly effective for studying biological sequences such as DNA, RNA, and proteins, uncovering patterns and relationships that are crucial for understanding biological functions and processes [5]. Furthermore, converting time series data to sequences can improve classification performance by utilizing effective sequence classification algorithms, which can be particularly useful in industrial monitoring to classify different machine states or detect early signs of equipment failure. This paper explores the novel idea of leveraging these powerful sequence analysis techniques [3] for time series data by transforming time series signals into representations similar to biological sequences.

The concept of transforming time series data into sequence-based representations is not entirely new. Previous research has explored various symbolic representation methods, such as Symbolic Aggregate approXimation (SAX) and Piecewise Aggregate Approximation (PAA), to simplify and discretize time series data [14, 18, 21]. These methods, however, often lack the ability to fully capture the complexity and rich information contained in the original time series [8]. Recent advances in representation learning and the success of k-mers-based representation in bioinformatics highlight the potential of adopting bioinformatics techniques for time series analysis [4]. The k-mers method, for example, segments sequences into overlapping substrings of length k, which can then be analyzed to uncover meaningful patterns and features [5].

In this paper, we propose a method to transform time series signals into biological/molecular sequence-type representations using a unique alphabetic mapping technique. By dividing the range of time series values into 26 distinct ranges, each corresponding to a letter of the English alphabet, we map each value in the time series to a specific character. This transformation results in a character sequence that retains the temporal ordering and relative magnitudes of the original time series data. The generated sequences can then be analyzed using sequence analysis algorithms from bioinformatics, such as k-mers-based analysis, which have been effectively used for identifying patterns in biological sequences. To demonstrate the effectiveness of our proposed method, we conducted experiments using a real-world time series dataset originally used for human activity recognition [2]. We transformed these datasets into character sequences and applied sequence classification and evaluated their performance. The results show that our approach not only simplifies the representation of time series data but also enhances classification accuracy by leveraging the robust sequence analysis techniques from bioinformatics.

The remainder of this paper is organized as follows: Section 2 discusses the previous literature. Section 3 provides a detailed description of the proposed method for transforming time series signals into sequence-type representations. Section 4 outlines the experimental setup and datasets used for evaluation. Section 5 presents the results of our experiments and discusses the implications of our findings. Finally, Section 6 concludes the paper and suggests potential directions for future research

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06

https://doi.org/XXXXXXX.XXXXXXX

2 LITERATURE REVIEW

The transformation of time series data into symbolic representations has been the subject of extensive research. Symbolic Aggregate approXimation (SAX) and Piecewise Aggregate Approximation (PAA) are among the most well-known methods for discretizing time series data. SAX reduces dimensionality and discretizes time series by mapping data points to a symbolic representation [18, 21]. PAA, on the other hand, divides the time series into equal-sized segments and calculates the mean value of each segment [14]. While these methods simplify and make the data more manageable, they often fail to capture the intricate patterns and rich information present in complex time series data [8].

Recent advancements in representation learning have opened new avenues for time series analysis. Representation learning aims to automatically discover the representations or features required for a specific task. Methods such as k-mers-based representation, which originated in bioinformatics for analyzing biological sequences, have shown promising results in uncovering meaningful patterns in time series data [4, 6]. The k-mers approach involves segmenting sequences into overlapping substrings of length k, enabling the capture of local sequence information and facilitating more effective pattern recognition [5, 10].

In addition to traditional symbolic representation methods, more sophisticated techniques have been developed to enhance time series classification and analysis. One such technique is the use of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to learn representations directly from raw time series data. These models have demonstrated significant improvements in classification accuracy and the ability to handle the non-linear dynamics of time series [11]. However, these methods often require large amounts of labeled data and substantial computational resources, making them less practical for certain applications [12].

The bioinformatics domain offers a rich set of sequence analysis tools and algorithms that can be leveraged for time series analysis. For example, sequence alignment algorithms, which are widely used for comparing DNA, RNA, and protein sequences, can be adapted to align and compare time series sequences [9]. Techniques such as hidden Markov models (HMMs) and dynamic time warping (DTW) have been successfully applied to time series data, demonstrating the potential of bioinformatics-inspired methods in this field [7, 17]. The adoption of these techniques for time series analysis enables the discovery of complex temporal patterns and relationships that are often missed by traditional methods [13].

3 PROPOSED APPROACH

In this section, we present the details of our proposed method that generates numeric representation from time series data, the flow of our approach is shown in Figure 1. Given a set of Time Series Signal data as shown in step (a) of Figure 1, we generate sequence embedding using Algorithm 1. The first step as seen in line number 3 of Algorithm 1 and step(b) of Fig 1, involves flattening the Time Series Signals data to assess and prepare the data for alphabetic mapping including finding the adequate range boundaries that divide the data into equal intervals in line number 4 of Algorithm 1 using function'ComputeRanges()' presented in Algorithm 2. This function uses the flattened data to find the global maximum and minimum values in the time series data as seen in lines 2 and 3 of Algorithm 2 and step (c) of Fig 1. In our case we choose 26 as the number of ranges in which we want the data to be divided which is in accordance with the total 26 Alphabets. This is followed by calculating the interval d of each range using the maximum value, minimum value, and the total number of ranges using the equation in line number 5 of Algorithm 2. Using this interval value we compute the bounds of all the ranges in line 11 of Algorithm 2 as depicted in step (d) of Figure 1.

Using these range boundaries and each signal from the original time series data we conduct the alphabetic mapping using Algorithm 3 by calling the function 'Mapping()' in line number 10 of Algorithm 1. As depicted by Algorithm 3 and the colored lines linked to step (g) of figure 1, this function at a time takes one signal as input and checks each value of the signal according to Range bounds to see which range that value lies in as shown in line 7 of Algorithm 3 and then based on the alphabet mapping rule observed in moving from step (d) to step (f) in Figure 1, each value belonging to a particular range is mapped to the respective alphabet in line 8 of Algorithm 3. As a result, we get a unique sequence of characters for each signal as depicted in step (h) of Figure 1 which is further used as input to 'Computekmers()' function in line number 11 of Algorithm 1 that works according to Algorithm 4 and extracts the k-mers from the sequence following the criteria shown in line 5 of Algorithm 4 and step (i) of Fig 1.

To finally generate the numeric embedding we use k-mers and their counts in each sequence as the numeric representation. For this, we represent each sequence as a spectrum that contains all possible k-mers that we get from line 7 of Algorithm 1 and compute their respective counts representing the number of times a particular kmer occurs in a sequence as shown in line 16 of Algorithm 1. The spectrum of each sequence is further appended in line number 18 of Algorithm 1 to form our final numeric representation of the time series data. These representations prove to be valuable for downstream tasks such as classification.

4 EXPERIMENTAL SETUP

We employed a standard 60-10-30% training, validation, and test set for our experiments, reporting the mean results across 100 iterations. To evaluate the classifiers' (i.e. Support vector machines(SVM), Naive Bayes (NB), Multilayer perceptron (MLP), K-nearest neighbors (KNN), Random Forest (RF), Logistic regression (LR), and Decision tree (DT)) performance, we utilized various metrics, including Average Accuracy, Precision, Recall, F1 (weighted), F1 (Macro), F1 (Micro), ROC AUC (one-vs-rest), and training runtime. As a baseline, we use the method proposed in [2].

Remark 1. *In our study, we chose not to use deep learning-based methods as baselines. This decision is primarily due to the data-hungry nature of deep learning models, which require large amounts of labeled data to perform effectively [11]. Many real-world time series datasets, particularly in specialized or emerging fields, often consist of limited data, making deep learning approaches less practical. Additionally, deep learning models tend to be computationally intensive, demanding significant resources for training and deployment [11, 12]. By focusing on symbolic and bioinformatics-inspired methods, we*

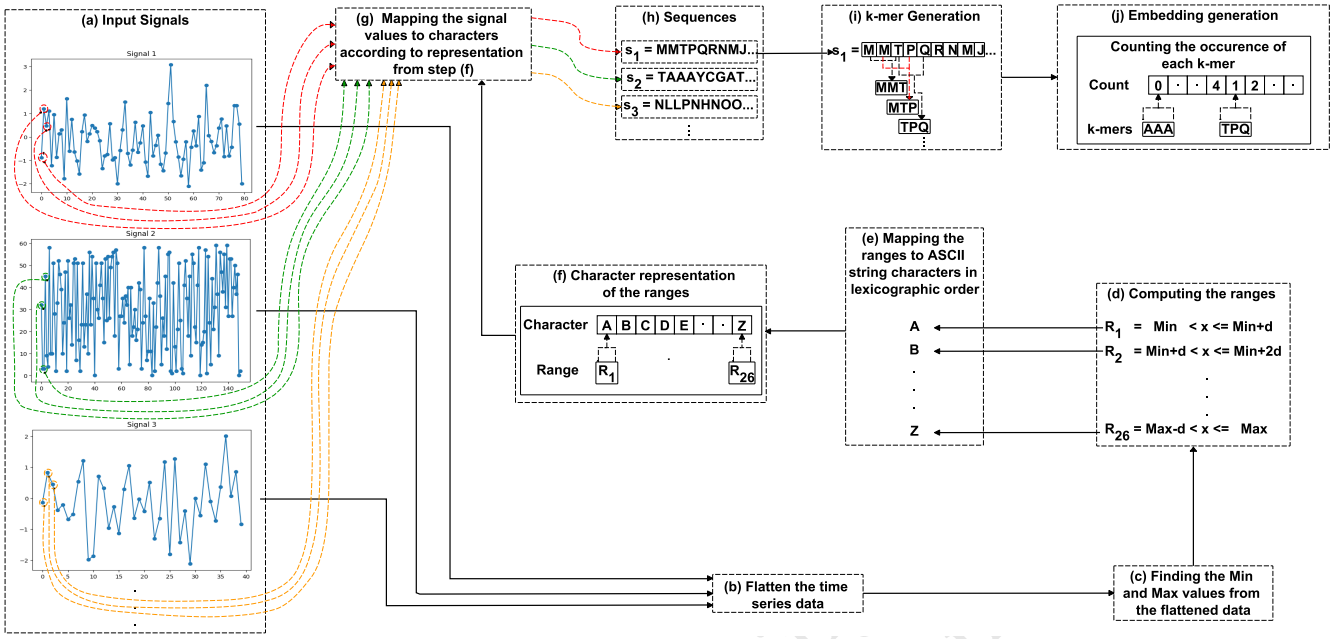

Figure 1: Flow diagram for the proposed approach

Algorithm 1 Sequence Embedding generation from Time Series Data

Input: Set of Time Series Signals (T), Alphabets (A)
Output: Sequence Embedding (ϕ)

function GenerateEmbedding(T)
  $\phi \leftarrow []$ ▷ Initialize an empty list
  FlatData ← Flatten(T) ▷ Flatten the time series signal data
  RangeBounds ← ComputeRanges(FlatData)
  Num ← Length(T) ▷ Number of Signals in input data
  kmersize ← 3 ▷ k-mer size
  $Globalkmers \leftarrow Combinations(A)$
  for j = 0 **to** Num **do**
    Signal ← T[j] ▷ Single Signal in the data
    Sequence ← Mapping(Signal, RangeBounds, A)
    Seqkmers ← Computekmers(Sequence)
    L = Length(Seqkmers)
    $kmersCount \leftarrow [0] * |A|^{kmersize}$
    for $kmer$ in Seqkmers **do**
      $idx = Globalkmers.index(kmer)$
      $kmersCount[idx] \leftarrow kmersCount[idx] + 1$
    end for
    $\phi.append(kmersCount)$
  end for
  return ϕ
end function

Algorithm 2 Computing Ranges

Input: Flattened Time Series Data (F)
Output: Boundaries of Ranges (B)

function ComputeRanges(F)
  MaxValue← Max(F) ▷ Finding the maximum value
  MinValue← Min(F) ▷ Finding the minimum value
  $NumRanges \leftarrow 26$ ▷ Number of ranges
  $d \leftarrow \frac{MaxValue - MinValue}{NumRanges}$ ▷ The interval in a range
  B ← [] ▷ Initialize an empty list
  for i = 0 **to** $NumRanges$ + 1 **do**
    if i=0 **then**
      Bound = F[i]
    else
      Bound = Bound + d ▷ Compute bounds of ranges
    end if
    B.append(Bound)
  end for
  return B
end function

aim to develop more accessible and resource-efficient techniques for time series analysis and classification.

For the dataset, we used the human activity smartphone sensor data from the accelerometer, magnetometer, and gyroscope as proposed in [2]. The dataset is originally collected from 29 users performing different human activities on their smartphones. The total number of data points (i.e. time series signals) is 112. As the target variable, we use the Gender of the users, the hand in which users were holding smartphones while performing different tasks, the application they were using on the smartphone, and the age of the users. The unique values for the age label of the 29 users are $<$ 20, 20-25, 25-30, 30-35, $>$ 35 while their respective count is 2, 7, 7, 6, and 7, respectively. The unique values for gender label are Male and Female while their respective distribution are 17 and 12,

Algorithm 3 Converting Time Series Data to Alphabetical Representation

Input: Time Series Signal(α), RangeBounds (R), Alphabets (A)
Output: Character Sequence (S)

```
function Mapping(α, R, A)
    NumValues = Length(α)            ▷ Number of Signal points
    S ← []                           ▷ Initialize an empty list
    for m = 0 to NumValues do
        Val = Signal[m]              ▷ Value of one signal point
        for i=0 to NumRanges do
            if R[i] <= Val < R[i+1] then
                Character = A[i]
            end if
        end for
        S.append(Character)
    end for
    return S
end function
```

Algorithm 4 Computing k-mers

Input: Character Sequence (C), ksize
Output: List of k-mers in a sequence (K)

```
function ComputeKmers(C, ksize)
    K ← []                           ▷ Initialize an empty list
    LenSeq ← Length(C)
    for i = 0 to LenSeq - ksize + 1 do
        kmer = C[i:i+ksize]          ▷ Extract k-mers from sequence
        K.append(kmer)
    end for
    return K
end function
```

respectively. The unique values for the application label are Facebook, Instagram, WhatsApp, and Twitter while their distributions are 28, 29, 28, and 27, respectively. The unique values for the Hand label are Left Handed, Right Handed, and Both hands (i.e. holding the smartphone with both hands) while their distributions are 10, 12, and 7, respectively.

5 RESULTS AND DISCUSSION

The results for the age prediction are shown in Table 1 for the baseline and our proposed method. We can observe that the proposed method outperforms the baseline for all metrics other than the training runtime. For the training runtime, since the embedding size of the baseline is smaller due to the usage of a smaller set of statistical features, their method can help ML models to train faster. However, in terms of predictive performance, the proposed method outperforms the baseline by 2.8% in terms of average accuracy and 1.9% in terms of ROC-AUC.

For the gender attribute, the maximum average accuracy and ROC-AUC that we got for the baseline are 0.982 and 0.980 using the SVM classifier. However, our proposed method's highest respective values are 0.990 and 0.991, which we got using the logistic regression classifier. Similarly, for the hand attribute, the maximum average accuracy and ROC-AUC that we got for the baseline are 0.434 and 0.563 using the decision tree classifier. However, our proposed method's highest respective values are 0.445 and 0.572, which we got using the decision tree classifier. Moreover, for the application attribute, the maximum average accuracy and ROC-AUC that we got for the baseline are 0.521 and 0.688 using the multi-layer perceptron classifier. However, our proposed method's highest respective values are 0.533 and 0.695, which we got using the logistic regression classifier. Note that we have not reported detailed results in tables for Gender, Application, and Hand attributes as we did for the Age attribute in Table 1 due to space limitations. Therefore, we only reported the best results for the baseline and proposed method for fair comparison. Moreover, it is noted that the proposed method consistently outperformed the baseline for all attributes in terms of all evaluation metrics other than classifier training runtime.

Table 1: Classification results (Averaged over 100 runs) on age prediction. The best values are shown in bold.

Method	ML Algo.	Acc.	Prec.	Recall	F1 weigh.	F1 Macro	ROC-AUC	Train. runtime (sec.)
Feature Engineering [2]	SVM	0.917	0.931	0.917	0.916	0.893	0.940	0.014
	NB	0.736	0.779	0.736	0.730	0.722	0.827	0.015
	MLP	0.925	0.938	0.925	0.923	0.905	0.948	0.227
	KNN	0.843	0.836	0.843	0.825	0.752	0.867	0.016
	RF	0.915	0.924	0.915	0.911	0.884	0.934	0.139
	LR	0.911	0.923	0.911	0.907	0.882	0.933	0.020
	DT	0.837	0.856	0.837	0.833	0.810	0.891	**0.009**
Ours	SVM	0.871	0.900	0.871	0.871	0.852	0.909	0.017
	NB	0.841	0.899	0.841	0.845	0.853	0.903	0.012
	MLP	0.894	0.917	0.894	0.894	0.892	0.937	0.318
	KNN	0.547	0.633	0.547	0.538	0.534	0.720	0.013
	RF	0.894	0.909	0.894	0.891	0.871	0.913	0.0127
	LR	**0.953**	**0.958**	**0.953**	**0.953**	**0.949**	**0.967**	0.033
	DT	0.888	0.904	0.888	0.886	0.821	0.899	0.010

Since we are running our experiments 100 times and reporting the average results, we also noted the standard deviations (SD) of 100 runs to observe the stability of the computed results. We noted that the SD values were very low in the majority of the cases, i.e. < 0.02, which showed that there is not much variation in the reported results. Moreover, we used the famous student t-test to evaluate the statistical significance of the classification results. Since the SD values were very low, the p-values were also < 0.05, hence showing that the reported results are statistically significant.

6 CONCLUSION

This paper presents a novel method for transforming time series signals into biological sequence-type representations using alphabetic mapping and k-mer strategy. We generated character sequences that retain the temporal ordering and relative magnitudes of the original data. This transformation enables the application of advanced sequence analysis techniques from the field of bioinformatics to time series data. Our experimental results demonstrate that the proposed method enhances classification accuracy. Future work could look into transformation for facilitating transfer learning from the bioinformatics domain, where powerful representation learning techniques, such as those used for DNA and protein sequence analysis, can be repurposed to improve time series analysis.

REFERENCES

[1] S Abilasha, Sahely Bhadra, Ahmed Zaheer Dadarkar, and P Deepak. 2022. Deep Extreme Mixture Model for Time Series Forecasting.. In *CIKM*. 1726–1735.

[2] Sarwan Ali. 2023. Information We Can Extract About a User from 'One Minute Mobile Application Usage'. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 1–6.

[3] Sarwan Ali and Murray Patterson. 2021. Spike2vec: An efficient and scalable embedding approach for covid-19 spike sequences. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 1533–1540.

[4] Sarwan Ali, Bikram Sahoo, Muhammad Asad Khan, Alexander Zelikovsky, Imdad Ullah Khan, and Murray Patterson. 2022. Efficient approximate kernel based spike sequence classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2022).

[5] Sarwan Ali, Bikram Sahoo, Naimat Ullah, Alexander Zelikovskiy, Murray Patterson, and Imdadullah Khan. 2021. A k-mer based approach for sars-cov-2 variant identification. In *Bioinformatics Research and Applications: 17th International Symposium, ISBRA 2021, Shenzhen, China, November 26–28, 2021, Proceedings 17*. Springer, 153–164.

[6] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology* 33, 8 (2015), 831–838.

[7] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*. 359–370.

[8] Xinye Chen. 2023. Joint symbolic aggregate approximation of time series. *arXiv preprint arXiv:2401.00109* (2023).

[9] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.

[10] Grant C Greenberg. 2023. *Analysis and applications of k-mer based methods in bioinformatics*. Ph. D. Dissertation. University of Illinois at Urbana-Champaign.

[11] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.

[12] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery* 34, 6 (2020), 1936–1962.

[13] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. 2011. Weighted dynamic time warping for time series classification. *Pattern recognition* 44, 9 (2011), 2231–2240.

[14] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* 3 (2001), 263–286.

[15] Wei-Lun Kuo, Tian-Shyr Dai, and Wei-Che Chang. 2021. Solving Unconverged Learning of Pairs Trading Strategies with Representation Labeling Mechanism.. In *CIKM Workshops*.

[16] Nijat Mehdiyev, Johannes Lahann, Andreas Emrich, David Enke, Peter Fettke, and Peter Loos. 2017. Time series classification using deep learning for process planning: A case from the process industry. *Procedia Computer Science* 114 (2017), 242–249.

[17] Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[18] Chotirat Ann Ralanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das. 2005. Mining time series data. *Data mining and knowledge discovery handbook* (2005), 1069–1103.

[19] Arijit Ukil, Leandro Marín, and Antonio J Jara. 2021. L1 and L2 Regularized Deep Residual Network Model for Automated Detection of Myocardial Infarction (Heart Attack) Using Electrocardiogram Signals.. In *CIKM Workshops*.

[20] Philip B Weerakody, Kok Wai Wong, Guanjin Wang, and Wendell Ela. 2021. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing* 441 (2021), 161–178.

[21] Byoung-Kee Yi and Christos Faloutsos. 2000. Fast time sequence indexing for arbitrary Lp norms. (2000).

[22] Zhi Zhang and Shenghua Wei. 2017. A Method for Short-Term Quantitative Precipitation Forecasting. CIKM.