BEHAVIOR ALGORITHM — A NOVEL TIME-SERIES CLUSTERING APPROACH

Shaho Alaee
George Washington University
ORCID: 0000-0002-4735-6552

Jason M. Pittman
University of Maryland Global Campus
ORCID: 0000-0002-5198-8157

Abstract- Clustering time-series values is an established technique for organizations employing machine learning to analyze temporal datasets. Generally speaking, the goal of time-series methodology is to generate predictions. Such predictions could help organizations understand potential future cyberattacks, financial market changes, weather, or disease outbreaks. However, computational limitations lead existing algorithms to fail to group individual series together based the actual behavior of the series. A feature that can be used or derived to explain the time-series behavior had not been identified in the literature despite there being a need to have numeric values to describe the pattern of values over time. To address this gap, this work presents a behavior algorithm which addresses clustering time-series data based solely on the behavior of the series. Further, the algorithm is designed to operate effectively regardless of absolute values or temporal shifts. First, we describe the algorithm through mathematical examples. We provide the design approach for the algorithm numerically and through data visualizations. Then, we validated the algorithm on sample random data. Finally, we offer conclusions along with notions for future work based on this study.

Keywords- time-series · clustering · behavior · algorithms

1. INTRODUCTION

A variety of industries such as cybersecurity [1], human performance [2], finance [3], and medicine [4] leverage time series analysis. The essential purpose of employing any form of time-series methodology is to mine a prediction out of a diverse dataset. These predictions have great value to society: cyberattacks, financial market predictions, weather forecasting, and disease outbreak management to mention a few timely examples. Given the social impact of time-series analysis, the value to science and technology is in the creation, implementation, and optimization of these algorithms[5].

A fundamental approach to analyzing such data is clustering [6], especially in the context of temporal data. However, existing algorithms for analyzing time-series data fail to quantify and visualize how different series compare based on temporal fluctuations [7, 8, 9]. Granted, fluctuations occur in most data samples and are unavoidable [10, 11]. Yet, this leaves a gap in which clustering categorical data by behavioral patterns is computationally expensive and does not provide insight beyond how a single series of data is related to another from a time-series perspective (e.g., a series that has high fluctuation or variation versus low).

This problem is significant to future forecasting efforts because such patterns aid in deeper understanding of the data, unforeseen similarities and differences across the dataset, clustering data points in a new dimension, and computational time savings when comparing time series data. Accordingly, this work addresses the lack of behaviorally adjusted time-series models by first identifying numeric features representing the behavior of the series. We then describe a novel algorithm aimed at improving the way time-series clustering and classification is conducted.

It is necessary to first establish the foundation related to this work. The next section offers a robust background of literature associated with time-series analysis. Particular areas of focus include research problems, the solutions offered throughout the history of the field, as well as synthesis of features across the research base important to the novel algorithm proposed in this study.

2. RELATED WORK

Time-series clustering is not new. Organizations base forecasting on time-series data through trend and seasonality to compute values such as rolling forecasts [12]. Standard forecasting techniques include analysis of trend and seasonality [13]. This approach is used to analyze a series and make predictions but does not provide a way to compare two series. For example, looking at a customer's expenditure and predicting their expenditures for the following year, or by grouping all customers together for a given country and predicting the following year. This led to the creation of other approaches with the goal of clustering time-series data. While many approaches currently exist in modeling and working with time-series data, a clustering method that looks at behavior did not [7, 8, 9, 14, 15, 16].

This section explores the conceptual framework of time-series analyzes. The conceptual framework emphasizes the strengths of four approaches representing the state of the art in terms of time-series clustering. To that end, Aghabozorgi, Seyed, and Wah [17] gathered time-series approaches that were used in the last decade. The authors identified the most common approaches as The Hausdorff distance, HMM-based distance, Dynamic Time Warping (DTW), Euclidean distance, short times series distance, and Longest Common Sub-Sequence (LCS).

In general, clustering techniques measure similarity of some sort and those which are closer in some space (i.e., a plane or vector) are more similar [18]. Each of the following algorithms also aim to measure similarity through distance, however with different approaches. Using distance as an approach does have success but exhibit exponential time complexity and thus increasing computational expense. This is due to the fundamental approach in how distance is measured.

First, all series have *x* amount of data points, where *x* is the number of values in the data set for a single series. Then, every series is measured against every other series. The output is a specific series or ID and a list of all other series or IDs that are similar based on proximity by measuring the distance. For instance, if a data set has ten different time-series, each with ten different time points with a corresponding value, all time-series clustering algorithms would take every series and measure the distance for use as a central vector for clustering. A trivial amount of data is not a problem; however, in real world applications the data set will become too large, rendering this approach infeasible. Additionally, this does not include the issue of the finding similar behavior between series', regardless of the values.

2.1 Euclidean Distance

Euclidean distance has high accuracy as it measures the distance between all the values [19]. Two limitations of this algorithm is that values need to be the same length (i.e., same dimensions) and the series need to be aligned temporally [17]. For example, imagine two customers spending money every month for one year. Measuring the Euclidean distance between the two would compare January to January, February to February, March to March, so on and so forth. This is okay if the goal is to compare expenditure with a fixed time variable. However, if the patterns were similar regardless of when expenditures happened, this approach wouldn't work. Put simply, calculating the Euclidean distance excels in finding series that have the same pattern at the same time points, when peaks and valleys happen at the same timestamps [14]. In contrast, when the series have non-matching fluctuations and do not match, it is very likely that their scores will indicate no similarity.

2.2 Dynamic Time Warping

DTW is an approach to taking time-series and finding similarities as opposed to identical time alignments as seen with Euclidean distance [20]. DTW was first created to identify similar sounds waves where high accuracy was important in trade-off of high time complexity [21]. Similar, non-music based problems can be handled by DTW as the algorithm deals well with time-series clustering in the face of temporal shifts [17] and works well on small data sets. However, the algorithm is limited by its space and time complexities [22, 23]. The reason behind the extensive computation is that every timestamp in the series has the difference measured with every timestamp of every other series. Different DTW approaches have been researched to overcome the challenge of computational limitations on large data sets [20]. Finally, DTW requires all series in the data set to be of equal sample size [17]. Consequently, when an exact match is not as vital based on the data set the computation limitations may not be worth the output or feasible depending on the application. Improvements have been made to reduce the time complexity, such as Mini-DTW, which aims to summarize a dataset so that it isn't as large [23]. However, this doesn't solve the gap of find similar patterns across different time-series without comparison.

2.3 Short Time-Series

The short time-series distance approach to clustering accounts for shorter temporal lengths in the data [17, 18]. This time-series algorithm clusters exact behaviors based on the time-series shape [24]. Such an approach is limited to clustering short time-series and exhibiting the same patterns [14]. Furthermore, short time-series cannot group different patterns together exhibiting the same behavior or fluctuations [17, 18]. At the same time, short time-series is resilient to absolute values and will not skew the data behavior if such are included in the input.

2.4 Level Shift Detection

Level Shift Detection (LSD) is an established time-series algorithm applicable to datasets containing a nontrivial quantity of outliers and anomalies [25]. LSD is applied in one of two forms: specific frequencies of patterns greater than a threshold or to find when the average over time changes in different segments of the series [26]. Similar series can be clustered by counting the number of occurrences for specific patterns [17, 14]. When looking at the second application where averages in certain windows change, LSD aims to measure how a series adjusts over time [27], not necessarily to cluster similar series together.

3. THE BEHAVIOR ALGORITHM

Current approaches to clustering focus on absolute values, which group series together where values are more similar [17]. These clustering mechanisms ignore the behavior of the series. Thus, the first step in using the behavior algorithm is to compute the sum of absolute difference and sum of series approach, after the data is normalized. It should be noted that using absolute values could cause large differences in distance measures depending on the variation of values that lie in the data set. Normalizing the data could find similarities of behavior by taking the effects of absolute values out of the equation regardless of the algorithm used. This was also demonstrated by [28] where taking the Euclidean distance of normalized data outperforms other algorithms and itself on absolute values.

The following (Figure 1) is an example of how absolute values and some standard descriptive statistics fail to identify patterns in time-series data. Series A has a value of 48,000 once, while series B has a value of 4,000 twelve times. The series averages and totals would be equal except the behavior was different between series A and B. The failure to identify patterns is caused by calculating the average and total amount and using them as metrics for distinction. It is not possible to conclude a difference in the behavior with such statistics.

Figure 1: The discrepancy based on fluctuation behavior in time-series data

Standard deviation and the variance outline the fluctuation behavior in the time-series behavior (Table 1). However, if standard deviation and variance can identify changes in the series behavior, then a new algorithm is not needed to group similar customers. Importantly, standard deviation and variance do not account for significantly different absolute values.

Table 1: Time-Series Descriptive Statistics

	Series A	**Series B**
Example	48,000 x 1	4,000 x 12
Average	4,000	4,000
Total	48,000	48,000
Std Dev	13,856	0
Variation	192,000,000	0

To extend the previous example demonstrating why absolute values will inaccurately handle fluctuation behaviors, we can visualize Series A with a value of 48,000 once compared to Series C with a value of 15,000,000 once (Figure 2). This example demonstrates two series where the behavior is the same but the standard deviation and variance are different. The visualization also shows behavior that is not easy to distinguish when plotted on the same graph with absolute values.

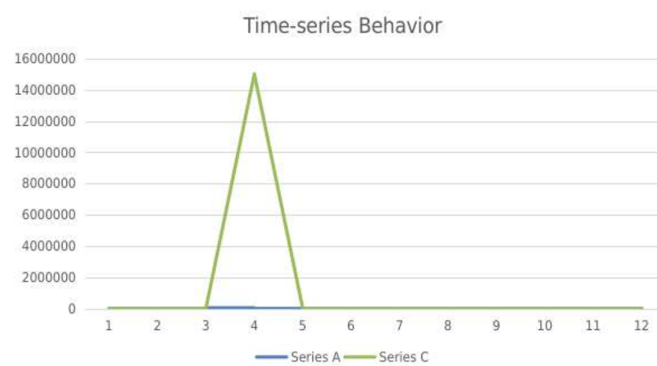

Figure 2: Visualizing hard to distinguish time-series behavior

Table 2: Descriptive Statistics for Two-Series Behavior

	Series A	**Series C**
Example	48,000 once	15,000,000 once
Average	4,000	1,250,000
Total	48,000	15,000,000
Std Dev	13,856	4,145,781
Variation	192,000,000	17,187,500,000,000

Based on the absolute values, series A and B had more in common because the average and total were the same. However, the actual behaviors of series A and C were more alike. Accordingly, the behavior algorithm must handle this discrepancy by normalizing the data first either by computing the traditional normalization technique or standard score (z-score). Further, absolute values make it difficult to group data elements based on their behavior which must beaccounted algorithmically.

For clarity, we take normalization to be a process of converting all values in a data list to values between zero and one. The normalization equation can be expressed as:

$$N_i = (X_i - X_{min})/(X_{max} - X_{min}) \qquad (1)$$

The z-score is a way to determine how many standard deviations (σ) a specific data point is from the mean of the dataset. The standardization (z-score) equation can be expressed as:

$$Z_i = (X_i - ,)/\sigma \qquad (2)$$

Table 3: Descriptive Statistics for Temporal Shifts

	Series A			Series B		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	21.00	1.00	0.00	111.00	1.00	0.00
STD	2.49	0.28	1.00	27.36	0.28	1.00
VAR	6.19	0.08	1.00	748.69	0.08	1.00
Mean	1.75	0.08	0.00	9.25	0.08	0.00
Median	1.00	0.00	-0.30	1.00	0.00	-0.30
Sum of Abs Diff	9.00	1.00	3.62	99.00	1.00	3.62

To account for the issue where the standard deviation is zero ($\sigma = 0$), the z-score equation can be modified to include epsilon (ϵ). Epsilon is a very small constant, much smaller than any meaningful standard deviation, to avoid a *division by zero* error. The modified standardization (z-score) with epsilon equation can be expressed as:

$$Z_i = (X_i - ,)/(\sigma + \epsilon) \qquad (3)$$

Using either approach places every data element on the same scale. Notably, the ϵ modified approach handled a large difference in the total amount between series leading to concomitant difference in the standard deviation and variances. Such large differences render clustering difficult, if not impossible. This becomes more evident when all three-series are plotted with standard normalized values (Figure 3).

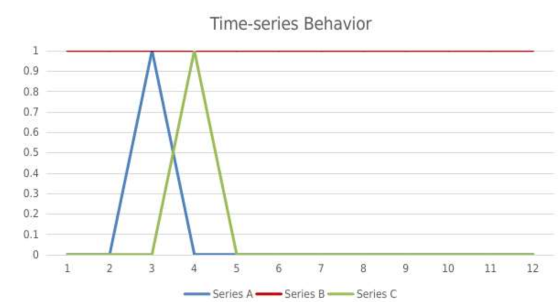

Figure 3: Visualizing clustering difficulties with three-series normalized values.

Series A and C are shifted to different time points for visibility but otherwise are identical. This data illustrates the two challenges of (a) identifying a way to handle the behavior of series regardless of absolute value; (b) and the difference of time (i.e., temporal shifts) serving as a confounding factor. Visually, we can see that normalization accounts for absolute values and acts a key step in the behavior algorithm's computation of the sum of absolute difference and sum of the series.

Series A and B (Table 3) both have values of one for eleven out of twelve timestamps. Series A has one value of ten whereas series B has one value of one hundred. The behavior is the identical. However, looking at the standard statistics, it is difficult to find the similarity. The sum of absolute difference and sum of series of the z-scores and normalization are the same for both series. Variance and standard deviation on the normalized data also appear to correctly identify the similarity in the pattern.

Table 4: Descriptive Statistics for Uneven Series Data

	Series C			Series D		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	120.00	12.00	0.00	1200.00	12.00	0.00
STD	0.00	0.00	0.00	0.00	0.00	0.00
VAR	0.00	0.00	0.00	0.00	0.00	0.00
Mean	10.00	1.00	0.00	100.00	1.00	0.00
Median	10.00	1.00	0.00	100.00	1.00	0.00
Sum of Abs Diff	0.00	11.00	0.00	0.00	11.00	0.00

Series C and D (Table 4) both have the same values for every point out of twelve in their respective series. Series C has twelve values of ten whereas series D has twelve values of one hundred. The behavior is the same, however looking at the standard statistics, the similarity is obscured. The sum of absolute difference and sum of series of the z-scores and normalization are the same for both series. Variance and standard deviation on the normalized data also appear to correctly identify the similarity in the pattern.

Table 5: Descriptive Statistics for Dissimlar Time-Series

	Series E			Series F		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	48.00	6.00	0.00	240.00	6.00	0.00
STD	1.63	0.41	1.00	8.16	0.41	1.00
VAR	2.67	0.17	1.00	66.67	0.17	1.00
Mean	4.00	0.50	0.00	20.00	0.50	0.00
Median	4.00	0.50	0.00	20.00	0.50	0.00
Sum of Abs Diff	30.00	7.50	18.37	150.00	7.50	18.37

Series E and F (Table 5) fluctuate between three different values. Both have each value appear four different times in a specific order following the pattern *123123123123*. The behavior is the same, however looking at the standard statistics, it is again difficult to find the similarity. The sum of absolute difference and sum of series of the z-scores and normalization are the same for both series. Variance and standard deviation on the normalized data also appear to correctly identify the similarity in the pattern.

To demonstrate that sum of absolute difference and sum of series do not just consider the values at the time points, whether absolute or normalized, another example is provided (Table 6). Series G and H are two series which have the same values but were recorded in a different pattern.

Table 6: Descriptive Statistics for Pattern Variance in Time-Series

	Series G			Series H		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	90.00	6.00	0.00	90.00	6.00	0.00
STD	2.50	0.50	1.00	2.50	0.50	1.00
VAR	6.25	0.25	1.00	6.25	0.25	1.00
Mean	7.50	0.50	0.00	7.50	0.50	0.00
Median	7.50	0.50	0.00	7.50	0.50	0.00
Sum of Abs Diff	55.00	11.00	22.00	5.00	1.00	2.00

Series G and H have the same values yet the pattern is different. Series G alternates between two values for every timestamp, whereas H has the same value for the first six timestamps. Then, Series H changes to a different value for the following six measures. Thus, the behavior is different. Standard statistics, such as standard deviation, variance, mean, and median, are identical. The sum of absolute difference of the z-scores and standard normalization express numerically the behavior is in fact different. However, the sum of series is the same for both as it does not account for the order of which the expenditure happened. The standard deviation and variance also identified similarity in patterns. Notably, when applied to a larger sample size, limitations start to arise.

To that end, Figure 4 visualizes the different data for series A (i.e., original, normalized, and z-score). A key takeaway from the visualization is the behavior of the series is still intact. These results point towards normalization being necessary either through traditional normalization or computing the z-score as the new scores do not alter the behaviorof the series.

4. Discussion

The z-score provides a type of normalization as well. A strength of the z-score is the sum of absolute difference adequately finds similarities in series where behaviors were identical. On the other hand, a limitation is summing the computed z-score (i.e., computing the sum of series), produces a total which is always zero due to the negative numbers. Therefore, the sum of series does not provide any level of distinction in those values. The standard deviation and variance calculated on the z-scores computed to one for every element, with a mean of zero. With variance and

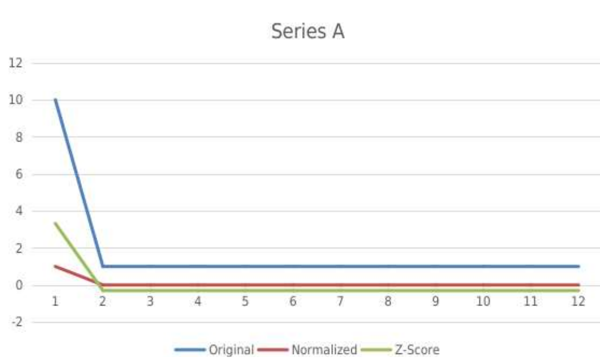

Figure 4: Visualizaing normalization of time-series data for time-series integrity.

standard deviation identified as attributes changing with high fluctuation in behavior, the z-score variance and standarddeviation lose their meaning.

The standard normalization's sum of absolute difference and sum of series also adequately compute scores that are reflective of their behavior and not absolute values. The additional benefit is that the sum of series also provides a level of distinction between different behaviors and a score that is the same for identical behaviors. The mean and median provide some insight as they also identify similarity in the behavior. These values could also match when behavior is significantly different due to how they are computed. Standard deviation and variance also provide a level of distinction between the different series in the fictional data set.

This highlights the importance of normalizing the data to find similarities in behavior. Normalizing the z-scores does turn the data into the exact values that normalizing the total amounts gets without the extra calculation of computing the average, standard deviation, to then compute the z-score so that it can be normalized to get the same results. It is recommended to normalize the actual values as the principle of parsimony dictates that the simpler model should be chosen when there is little benefit to the more complex one. In this case the outputs derived are exactly equal, thereforeforgoing the extra steps is best practice.

Furthermore, not only does the sum of absolute difference help prepare the series data to be clustered based on similarity, the values are interpretable. A value of zero, or close to zero, will occur when all values are the same or if the series has a small sample. A value of one suggests all the values were the same except for one. This can occur for any length of time-series. For example, a series with two data points will equal one as well as a series with three data points where two are the same value. When values are greater than one, the larger the sum of absolute difference, the closer the data points are together in the series. Series E and F demonstrate this point. The smaller the sum of absolute difference, the larger the gap the between the highest value and the rest. Series A and B demonstrate this point. Depending on the pattern, the sum of absolute difference can identify a specific behavior that the sum of series cannot. The sum of absolute difference unique example demonstrates this point (Figure 5).

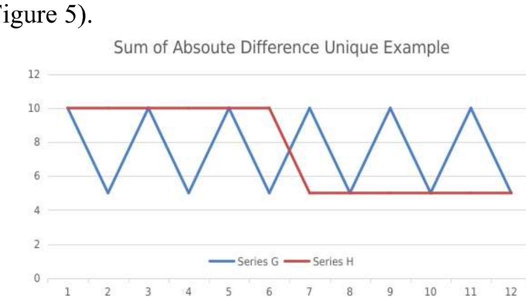

Figure 5: Visualizing patterns of specific behaviors based on absolute difference.

Here, Series G and H would have a sum of series value of six where no distinction could be made, whereas the sum of absolute difference would be one for series H and eleven for series G. Understanding this upfront can help determine which value is important to use and what the interpretation implies. The sum of series provides similar insight, however one aspect differs from sum of absolute difference. The sum of series exhibits the same behavior at both the minimum and maximum values (i.e., a drastic incline or decline). The values towards the middle exhibit a more regular behavior. The lower the sum, the more variation the series has while the higher the sum, the lower the variation until it reverses back to more variation (Figure 6).

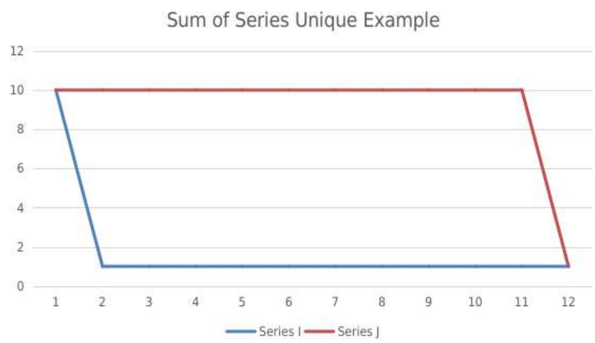

Figure 6: Visualizing variation in time-series with fluctuation behaviors.

The sum of absolute difference in both series would be the same- one- whereas the sum of series would be higher for series J and lower for series I. Series I would have a sum of series value of 1 and series J would have a sum of series value of eleven.

4.1 Interval Selection

Feature engineering is a key step to optimize time-series output for interpretability. To that end, a data preprocessing step can help regardless of the absolute values. The feature is interval selection. Depending on the appropriate choice, data then needs to be aggregated for that interval. For example, if daily data is available, but the desired interval is monthly, then the daily data must be summed for monthly aggregation. Selecting the appropriate period for analysis is critical. The aggregation period is relative to the application, but is nonetheless necessary. Figure 7 demonstrates data aggregation depending on the period selected. The time in the example is in calendar days.

Figure 7: Visualizing periodized data aggregation in time-series data.

Selecting the appropriate interval eliminates fluctuations while keeping the essence of the behavior. Looking at a daily interval may show a high fluctuation. Relatedly, the daily interval renders it difficult to interpret if the time-series is too long. Quarterly and yearly provide too little data points to fully capture the essence of the behavior. The patterns tend to appear similar with too much aggregation. Based on this example, the monthly interval gives the best description of behavior.

Further, the sum of absolute difference and sum of series will reveal patterns in the data. Thus, it is preferable to have too many data points rather than too few. For example, daily and monthly (Figure 7) have similarities based on the output while the pattern is difficult to see in daily visualizations. If the time points measured are less (e.g., quarterly and yearly), then the scores show similarities between different series that may not be present when visualized monthly. Of note, data over a longitudinal period (e.g., ten years) could yield deeper insight whereas monthly may be too noisy. This highlights the importance of selecting the correct interval based on the application.

4.2 Rounding

Rounding is another way to optimize time-series output. Similar items will still be clustered together based on the score, however rounding allows for a smoothness of the series and leads to more rigid values. In this way, the fluctuation evolves towards a more fluid pattern. To demonstrate (Figure 8), given twelve values between 1000 and 1005, the data were rounded to the nearest thousand. Doing so achieves the goal to smooth fluctuations in relatively similar values.

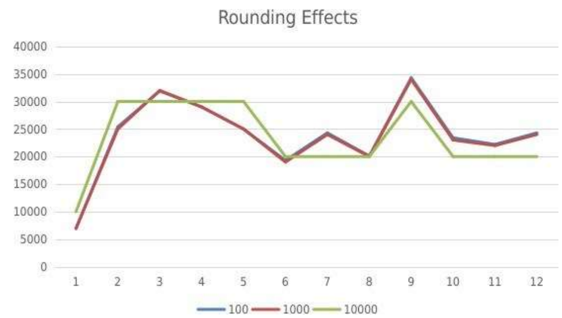

Figure 8: Visualization of smoothed time-series outputs.

Given the data set, when rounded to the nearest hundred and thousand the results are similar and does not add any level smoothness to the series. When rounding to the nearest ten thousand though, the variation becomes much less and causes the series to exhibit less fluctuation. This step can provide more robustness over the model to aid in visualizing patterns but must be completed with caution.

Rounding can cause time-series data, intended to be separated, to have similar scores. A sound approach then is rounding to specific places based on data features (e.g., rounding to the nearest ten for car speeds, hundred for plane speeds, and thousands for rocket speeds). As well, for clarity- rounding aids with visualization. Rounding is not necessary in computing the sum of absolute difference of sum of series.

5. CONCLUSION

Time-series analysis is a critical tool in a variety of industries. The chief use of time-series in these industries is to compute predictions based on a historical dataset. Such predictive capacity gives society the ability to predict diverse events such as cyberattacks, financial market changes, weather forecasting, and disease outbreaks. Further, clustering is heavily employed to aid in time-series data analysis, particularly when the data have a temporal attribute. However, existing algorithms for analyzing time-series data fail to quantify and visualize how different series compare based on temporal fluctuations [7, 8, 9]. Thus, a gap exists because clustering categorical data by behavioral patterns is computationally expensive and does not provide insight beyond how a single series of data is related to another from a time-series perspective (e.g., a series that has high fluctuation or variation versus low).

Identifying a new way to cluster time-series data was motivated by a need to overcome the existing, limited approaches. This included the creation of the behavior algorithm which computes the sum of absolute difference and sum of series features. Overall, the behavior algorithm demonstrates one potential solution to the research problem.

More specifically, the comparison of the absolute, normalized, and z-score values combined with computing the traditional statistics, sum of absolute difference, and sum of series, identified which values appropriately quantify the behavior of the series. Further, the sum of absolute difference and sum of series of the normalized data was able to distinguish the difference of data, even given the changes in absolute values. In this way, the sum of absolute difference and sum of series values overcome limitations in time-series clustering.

Lastly, providing a single value describing the behavior of the series and improving the computational complexity by eliminating the need to measure every data point (i.e., DTW), or by finding the difference between every time point in one series compared to another (i.e., Euclidean distance). Now with one value, clustering can be done in a fraction of the time, with very reliable results. This is achieved by computing the sum of absolute difference and sum of series for every series, and then either clustering all series together or using the values for classification using machine learningalgorithms.

With respect to future work, we first suggest work be done to produce an operational prototype implementing the time-series behavior algorithm. Of course, validation experiments would be necessary to quantify efficacy of the approach. Additional work then could be done to implement the behavior algorithm in a machine learning pipeline. Ex- perimentation in this area could develop a robust basis for understanding appropriate and inappropriate implementations across types of classifiers (i.e., supervised versus unsupervised). Finally, future work may be of interest in computational optimizing of behavior algorithm. Again, experimentation may reveal space and time complexity optimization with generalizability across types of time-series data.

6. REFERENCES

[1] Abdulkadir Bilen and Ahmet Bedri Özer. Cyber-attack method and perpetrator prediction using machine learning algorithms. *PeerJ Computer Science*, 7:e475, 2021.

[2] E Ramanujam, Thinagaran Perumal, and S Padmavathi. Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review. IEEE Sensors Journal, 21(12):13029–13040, 2021.

[3] Deepak Kumar, Pradeepta Kumar Sarangi, and Rajit Verma. A systematic review of stock market prediction using machine learning and statistical techniques. Materials Today: Proceedings, 2021.

[4] Mohammad Shehab, Laith Abualigah, Qusai Shambour, Muhannad A Abu-Hashem, Mohd Khaled Yousef Shambour, Ahmed Izzat Alsalibi, and Amir H Gandomi. Machine learning in medical applications: A review of state-of-the-art methods. Computers in Biology and Medicine, 145:105458, 2022.

[5] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. Econometric reviews, 29(5-6):594–621, 2010.

[6] D Neha and BM Vidyavathi. A survey on applications of data mining using clustering techniques. International Journal of Computer Applications, 126(2), 2015.

[7] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. International journal of forecasting, 22(3):443–473, 2006.

[8] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In European business intelligence summer school, pages 62–77. Springer, 2012.

[9] José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. Big Data, 9(1):3–21, 2021.

[10] Chong Li, Yingjie Yang, and Sifeng Liu. A new method to mitigate data fluctuations for time series prediction. Applied Mathematical Modelling, 65:390–407, 2019.

[11] Yang Lin, Irena Koprinska, and Mashud Rana. Springnet: Transformer and spring dtw for time series forecasting. In International Conference on Neural Information Processing, pages 616–628. Springer, 2020.

[12] Galit Shmueli and Kenneth C Lichtendahl Jr. Practical time series forecasting with r: A hands-on guide. Axelrod Schnall Publishers, 2016.

[13] Srihari Athiyarath, Mousumi Paul, and Srivatsa Krishnaswamy. A comparative study and analysis of time series forecasting techniques. SN Computer Science, 1(3):1–7, 2020.

[14] T Warren Liao. Clustering of time series data—a survey. Pattern recognition, 38(11):1857–1874, 2005.

[15] Ali Javed, Byung Suk Lee, and Donna M Rizzo. A benchmark study on time series clustering. Machine Learning with Applications, 1:100001, 2020.

[16] Ali Alqahtani, Mohammed Ali, Xianghua Xie, and Mark W Jones. Deep time-series clustering: A review.
Electronics, 10(23):3001, 2021.

[17] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review.
Information systems, 53:16–38, 2015.

[18] Jorge Caiado, Elizabeth A Maharaj, and Pierpaolo D'Urso. Time series clustering. Handbook of cluster analysis, pages 241–263, 2015.

[19] Ville Hautamaki, Pekka Nykanen, and Pasi Franti. Time-series clustering by approximate prototypes. In 2008 19th International conference on pattern recognition, pages 1–4. IEEE, 2008.

[20] Pavel Senin. Dynamic time warping algorithm review. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, 855(1-23):40, 2008.

[21] Meinard Müller. Dynamic time warping. Information retrieval for music and motion, pages 69–84, 2007.

[22] Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(6):623–635, 1980.

[23] Phudit Thanakulkairid, Tanupat Trakulthongchai, Naruesorn Prabpon, and Pat Vatiwutipong. Efficiency of time series clustering method based on distribution of difference using several distances. In 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), pages 1–6, 2022.

[24] Michio Sugeno and Takahiro Yasukawa. A fuzzy-logic-based approach to qualitative modeling. IEEE Transactions on fuzzy systems, 1(1):7–31, 1993.

[25] Ruey S Tsay. Outliers, level shifts, and variance changes in time series. Journal of forecasting, 7(1):1–20, 1988.

[26] Nathan S Balke. Detecting level shifts in time series. Journal of Business & Economic Statistics, 11(1):81–92, 1993.

[27] Nathan S. Balke. Detecting level shifts in time series. Journal of Business Economic Statistics, 11(1):81–92, 1993.

[28] Michael R Berthold and Frank Höppner. On clustering time series using euclidean distance and pearson correlation.
arXiv preprint arXiv:1601.02213, 2016.