An Approach to Time Series Classification Using Binary Distribution Tree

Chao Ma
School of Cyber Science and Engineering
Wuhan University
Wuhan, P.R.China
Email: chaoma@whu.edu.cn

Xiaochuan Shi
School of Cyber Science and Engineering
Wuhan University
Wuhan, P.R.China
Email: shixiaochuan@whu.edu.cn

Weiping Zhu*
School of Computer Science
Wuhan University
Wuhan, P.R.China
Email: wpzhu@whu.edu.cn

Wei Li
School of Cyber Science and Engineering
Wuhan University
Wuhan, P.R.China
Email: $auto_weili@whu.edu.cn$

Xiaohui Cui
School of Cyber Science and Engineering
Wuhan University
Wuhan, P.R.China
Email: xcui@whu.edu.cn

Hao Gui
School of Computer Science
Wuhan University
Wuhan, P.R.China
Email: hgui@whu.edu.cn

***Abstract*—As a typical task of time series mining, Time Series Classification (TSC) has attracted lots of attention from both researchers and domain experts due to its broad applications. To get rid of costly hand-crafting feature engineering process, deep learning techniques are applied for automatic feature extraction, which shows competitive or even better performance compared with state-of-the-art TSC solutions. However, on time series datasets presenting complex patterns, neither 1-Nearest-Neighbour classifier nor deep learning models are capable of achieving satisfactory classification accuracy which motivates us to explore new time series representations to help classifiers further improve the classification accuracy. In this paper, by building the binary distribution tree, an approach to time series classification based on deep learning models using new representations is proposed. By conducting comprehensive experiments over 6 most challenging time series datasets and comparing experimental results of the same classifier using the proposed representation or not, the potential of the proposed approach to enhancing time series classification accuracy is validated with a bunch of helpful findings.**

***Keywords*-time series classification; representation; binary distribution tree; deep learning;**

I. INTRODUCTION

In the past decade, the time series data are generated from various domains at a rapid speed. As a typical task of time series mining, Time Series Classification (TSC) has attracted lots of attention from both researchers and domain experts. To solve TSC problems, 1-Nearest-Neighbour classifier based on Euclidean distance [2] is usually selected as a baseline. Besides, many feature-based [4] [6] [7] [15] and bag-of-pattern based [9] [10] [8] [11] TSC solutions are proposed with various strategies for extracting discriminative features. However, domain knowledge is usually required for extracting significant features which is time-consuming and varies much with experts. Recently, therefore, deep learning models are applied to TSC problems [13].

*Corresponding Author Email: wpzhu@whu.edu.cn

Although there are so many TSC solutions, on datasets presenting complex patterns, most of existing TSC solutions can not achieve satisfactory accuracy. This may indicate that the limited accuracy is probably caused by the inappropriate representation of time series rather than the classifier. Hence, we propose a new time series representation based on the binary distribution tree with the hope that it can help existing TSC classifiers for further improving the classification accuracy.

The rest part of this paper is organized as follows. In Section II, the methodology is illustrated in detail. In Section III, the comprehensive experiments are conducted on 6 challenging datasets with thorough analysis about the impact of main factors. Related works about existing TSC solutions are discussed in Section IV followed with the conclusions and future works offered in Section V.

II. METHODOLOGY

A. Problem Formulation

The TSC problem can be formulated as follows.

Given:

- A set of all class labels $C = \{c_k\}(1 \le k \le K)$;
- A train set of time series $TS_{train} = \{ts^p_{train}\}(1 \le p \le P)$ in which each time series ts^p_{train} is attached with one class label $c_r \in C$;
- A test set of time series $TS_{test} = \{ts^q_{test}\}(1 \le q \le Q)$.

Assume:

- Each time series $ts = \{tpv_i\}(1 \le i \le N)$ is consisting with a set of consecutive numerical values (i.e the time point value tpv_i must be a real number);
- The class label of each time series $ts^q_{test} \in TS_{test}$ must be included in the class label set C;
- The class labels of time series in TS_{test} are unknown before classification but available after classification.

Objective:

978-1-7281-5212-7/19/$31.00 ©2019 IEEE
DOI 10.1109/MSN48538.2019.00082

- Maximize the classification accuracy Acc_F^R defined as follows:

$$Acc_F^R = \frac{\sum_{q=1}^{Q} hit_q}{Q} \tag{1}$$

where hit_q is 1 if the class label of the time series $ts_{test}^q \in TS_{test}$ is correctly predicted by the classifier F using representation R, otherwise hit_q is 0.

B. Representation using Binary Distribution Tree

In this paper, an approach using Binary Distribution Tree (BDT) to representing time series is proposed. The entire procedure for obtaining the BDT-based representation is consisting with three main stages: Binary Subsequence Tree (BST) construction, Binary Distribution Tree (BDT) construction and representation generation. More details about BST construction on time series ts with the split ratio sr are provided as shown in Algorithm 1.

Algorithm 1 Build_BST

Input: ts, sr in
Output: $BST(ts)$ out
Initialisation:
$BST(ts)_{root}^{root} = ts$
if $(len(ts) == 1)$ **then**
  return $BST(ts)$
end if
Iterative Process:
$sp = integer(len(ts) * sr)$
if $(sp == 1)$ **then**
  return $BST(ts)$
end if
$left\_series = ts[1, sp]$
$right\_series = ts[sp + 1, len(ts)]$
$BST(ts)_{root}^{left} = Build\_BST(left\_series, sr)$
$BST(ts)_{root}^{right} = Build\_BST(right\_series, sr)$
Output Binary Subsequence Tree for ts:
return $BST(ts)$

Once the construction process of BST is done, the second stage could be launched for generating the Binary Distribution Tree for ts. Given $BST(ts)$, TS_{train} and $bins$, the way for constructing the binary distribution tree $BDT(ts)$ for ts is shown as shown in Algorithm 2.

When the above two stages are finished, the last stage to obtain the BDT-based representation of ts is to concatenate the distribution of each node at the level l of $BDT(ts)$ as a vector. Therefore, for the BDT-based representation, there are 3 parameters (i.e. sr, $bins$ and l) to uniquely determine a specific representation which are shown in Table I. By executing all above 3 steps on all original time series from both train and test sets, the original time series are mapped from the raw representation space to BDT-based representation space and can be fed into classifiers for classification.

Algorithm 2 Build_BDT

Input: $BST(ts), TS_{train}, bins$ in
Output: $BDT(ts)$ out
Initialisation:
$minimum = argmin(TS_{train})$
$maximum = argmax(TS_{train})$
$bin\_width = (maximum - minimum)/bins$
Determining Bin Edges:
$bin\_edges[1] = minimum$
for $i = 1$ to $bins$ **do**
  $bin\_edges[i + 1] = bin\_edges[i] + bin\_width$
end for
Transforming $BST(ts)$ into Distribution Space:
for $eachnode$ in $BST(ts)$ **do**
  $BDT(ts)_{node} = histogram(BST(ts)_{node}, bin\_edges)$
end for
Output Binary Distribution Tree for ts:
return $BDT(ts)$

Symbol	Description	Range
sr	Split ratio for determining the split position	(0.00, 1.00)
$bins$	Number of bins for calculating bin edges	(1, ∞)
l	Level of BDT nodes for generating representation	[0, L)

Table I
PARAMETERS OF BDT-BASED REPRESENTATION

C. Classifier Structure

As the representation of time series being ready, the next step is to select proper classifier. In this paper, 1-Nearest-Neighbour classifier based on Euclidean distance is selected because it is of simple implementation and need no parameter tuning. Therefore, the difference on classification performance by adopting 1NN classifier using the proposed BDT-based representation or not can be easily identified.

Since being attracted by the promising potential of deep learning models for solving TSC problems [13], we are interested in recruiting 3 deep learning models Multi-Layer Perceptron (MLP), Fully Convolutional Network (FCN) and Deep Residual Network (ResNet) as the candidate classifiers for validating the effectiveness of the BDT-based representation. The structure and hyper-parameter settings of MLP, FCN and ResNet are shown in Figure 1.

III. EVALUATION

A. Experimental Settings

For validating TSC solutions, the most widely used time series datasets collected from different application fields

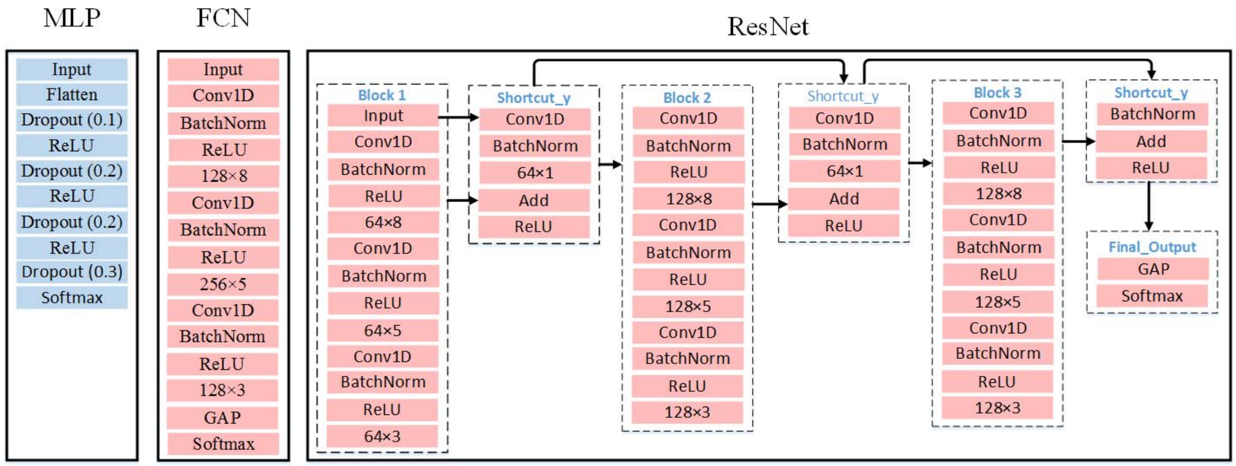

Figure 1. Structure and Hyper-parameters of Deep Learning Classifiers

Name	Classes	Train/Test	Length
Phoneme	39	214/1896	1024
Haptics	5	155/308	1092
InlineSkate	7	100/550	1882
InsectWS	11	220/1980	256
Herring	2	64/64	512
ScreenType	3	375/375	720

InsectWingbeatSound is denoted as InsectWS for short.

Table II
DATA SETS DESCRIPTION

are archived by [16] called UCR Archive. Although there are more than 80 datasets in UCR Archive, we select 6 most challenging datasets on which neither 1NN Euclidean classifier nor deep learning models are able to achieve satisfactory classification accuracy. More details about these 6 datasets are provided in Table. II.

As explained in Table I, there are three parameters (i.e. sr, $bins$ and l) to uniquely determine a BDT-based time series representation. In our experiments, the value of sr is initialized as 0.1 and increased by 0.1 until it reaches 0.9. And $bins$ is initialized as 3 and increased by 1 until it reaches 11. l is set to 1 at first and increased by 1 until it reaches 9. Since the output of deep learning models are not deterministic, we have run 10 iterations and 100 epochs for each deep learning classifier with the same representation and take the average accuracy as their final performance. One more thing to note is that the accuracy of the deep learning classifiers on test set is taken at the epoch when the loss on the train set reaches the lowest.

B. Overall Results and Analysis

The experimental results of 4 selected classifiers over 6 challenging datasets with or without the BDT-based representation are shown in Table III. Obviously, over all 6 datasets, 1NN Euclidean classifier using BDT-based representation outperforms its counter-part with the raw representation and the classification accuracy is significantly enhanced from 0.377 to 0.456 in average.

For MLP, BDT-based representation helps it obtain performance improvement on all 6 datasets and the average accuracy is enhanced by 5.8%. When we check results of ResNet and FCN, we see that on 5 out of 6 datasets the TSC accuracy is significantly improved while the average accuracy of both is boosting to 0.484 and 0.458 respectively. And absolutely it can be confirmed that the accuracy enhancement is brought by the BDT-based representation because each classifier using any representation is of exactly the same configuration.

Since each BDT-based representation is determined by three parameters sr, $bins$ and l, their impact on TSC accuracy for each selected classifier are investigated separately in Sections III-C, III-D and III-E.

C. Impact of Split Ratio on TSC Accuracy

To investigate the impact of split ratio sr on classification accuracy over selected datasets, the average accuracy of all the 4 classifiers on each dataset across 9 different values of sr is shown in Figure 2. For all datasets, obviously, they tend to obtain higher classification accuracy when the value of split ratio sr sitting in the range between 0.4 and 0.7.

D. Impact of Bin Number on TSC Accuracy

For the parameter $bins$, intuitively, the too small $bins$ tends to make the BDT-based representation become undiscriminating among times series especially with many categories while the too large $bins$ will make the BDT-based representation being too sensitive to the noisy data points in time series. To study the impact of $bins$ on classification accuracy, experiments are conducted and detailed results are

Classifier	Representation	Haptics	Herring	InlineSkate	InsectWS	Phoneme	ScreenType	Avg. Acc.
1NN Euc	RAW	0.37	0.516	0.342	0.562	0.109	0.36	0.377
	BDT	**0.435**	**0.75**	**0.413**	**0.573**	**0.161**	**0.461**	**0.466**
MLP	RAW	0.419	0.594	0.336	0.618	0.087	0.397	0.409
	BDT	**0.468**	**0.734**	**0.404**	**0.623**	**0.19**	**0.477**	**0.483**
ResNet	RAW	0.377	0.625	0.187	0.505	**0.319**	**0.605**	0.436
	BDT	**0.445**	**0.734**	**0.413**	**0.568**	0.249	0.475	**0.481**
FCN	RAW	0.334	0.422	0.187	0.244	**0.249**	**0.619**	0.343
	BDT	**0.432**	**0.719**	**0.405**	**0.516**	0.226	0.464	**0.46**

Table III
OVERALL TIME SERIES CLASSIFICATION RESULTS

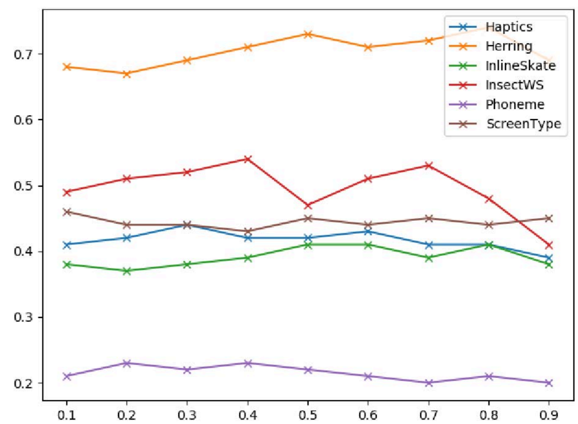

Figure 2. Impact of Split Ratio on TSC Accuracy

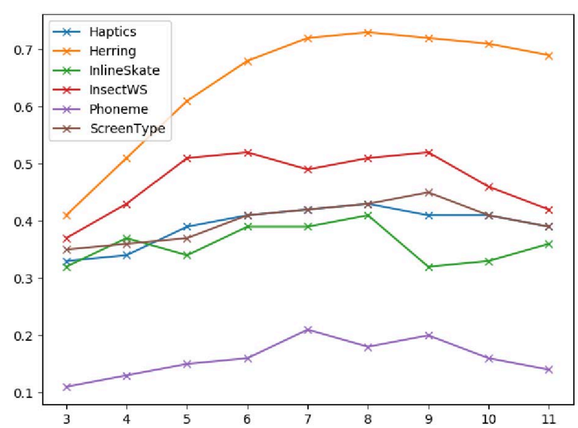

Figure 3. Impact of Bin Number on TSC Accuracy

shown in Figure 3. And it is clearly seen that on almost all tested datasets benefit from the larger value of $bins$ to different extents. And when $bins$ becomes larger than a certain value, either the best accuracy or the average accuracy tends to fluctuate or deteriorate.

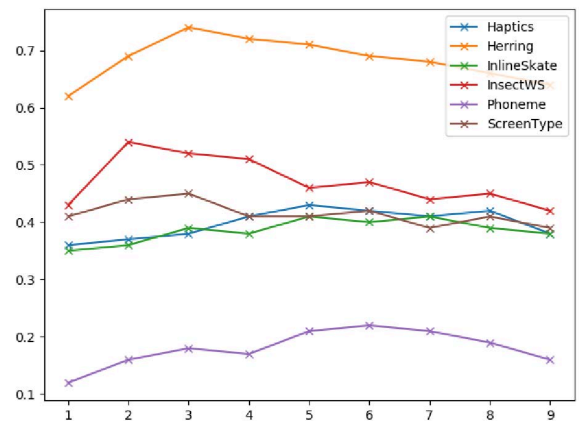

Figure 4. Impact of BDT Level on TSC Accuracy

E. Impact of BDT Level on TSC Accuracy

To obtain the final representation of time series, the parameter l should be set to indicate the level of BDT nodes for concatenating their values (distribution) to a vector as the new time series representation. Therefore, the greater the $level$ is, the more informative the BDT-based representation would be. The experimental results to help us investigate the impact of l over classification accuracy are shown in Figure 4. We find that datasets with shorter series such as InsectWS, Herring and ScreenType benefit more from smaller value of $level$ while it is reversed for the rest of datasets.

IV. RELATED WORK

For solving TSC problems, the most frequently adopted baseline is probably the 1-Nearest-Neighbour classifier based on Euclidean distance (denoted as 1NN-EUC for short) due to its parametric-free and time-efficient features [2]. Then, inspired by the success of Dynamic Time Warping technique in speech recognition, 1-Nearest-Neighbour Dynamic Time Warping (denoted as 1NN-DTW for short) is introduced into time series analysis [1].

Different from 1NN-EUC and 1NN-DTW which take the whole series similarity into account, feature-based TS classifiers are constructed on the basis of either local shapelets or bag-of-patterns (BOP) of time series. Shapelets are usually defined as a set of subsequences of the original time series, which is regarded as the most discriminative features for classifying time series. According to the recent evaluation on existing TSC solutions [2], Shapelet Transform (ST) [4] [6] is regarded as the most accurate shapelet-based method. In [7], a decision tree is built based on the distance to a set of shapelets. And the Learning Shaplets (LS) approach is proposed in [15] which generates optimal shapelets in the synthetic manner. However, the computation cost of the shapelet-based methods is always the main concern which limits their application in TSC.

For the BOP branch of feature-based TS solutions, Symbolic Aggregate approXimation (SAX) is regarded as the first published BOP approach, which transform the raw time series into a sequence of characters by using the sliding window with fixed length and then employ the 1NN classifier based on the self-defined distance between two character sequences for classification [9]. SAX is extended by SAX-VSM [10] by combining tf-idf weighted features with Cosine distance to obtain the only feature vector for each class, which not only saves memory space but also speeds up the execution time. TS Bag-of-features Framework (TSBF) [8] is another member of BOP family. The idea of TSBF is to build a supervised codebook generated by the random forest classifier with windows at random positions with random lengths. The BOP-based model BOSS (Bag-of-SFA-Symbols) [11] is currently the most accurate BOP-based approach by replacing SAX with Symbolic Fourier Approximation (SFA) [12].

To further enhance the accuracy, ensemble solutions are designed by incorporating different types of core classifiers and making the final decisions based on techniques such as majority voting, bagging or weighted aggregation. Elastic Ensemble (EE) [5] employees 11 core classifiers while COTE [3] combines 35 core classifiers including EE. The performance of the ensemble TSC solutions depend much on the variety of core classifiers and the decision making strategy.

Recently, there are attempts on applying deep learning models to solving TSC problems. In [13], for instance, several types of deep learning models are directly applied to classifying time series without any preprocessing. According to the experimental evaluation in [13], FCN performs the best among all tested deep learning models which shows competitive performance to COTE and BOSS.

V. Conclusion and Future Works

In this paper, a new time series representation based on binary distribution tree is proposed. By conducting experiments for 4 classifiers using raw representation or using the proposed representation over 6 challenging data sets, the potential of the proposed representation for further enhancing the classification accuracy is validated. Furthermore, comprehensive analysis about the impact of all factors (i.e. split ratio, bins number and representation level) is performed to offer deep insights for selecting proper parameters. And as it is mentioned in Section II that the BDT-based representation could be integrated with any classifier which is able to handle vectorized representations. In the future, therefore, more classifiers with more possible BDT-based representations are planned to be tested for exploring the potential of this new time series representation in solving TSC problems.

Acknowledgment

The authors would like to thank Hanqing Wu for offering the GPU server which greatly speeds up the experimental process. This research is supported in part by National Key R&D Program of China No.2018YFC1604000, National Natural Science Foundation of China No.61502351, and Chutian Scholars Program of Hubei.

References

[1] Donald J. Berndt and JamesClifford. Using dynamic time warping to find patterns in time series. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAIWS'94, pages 359–370, AAAI Press, 1994.

[2] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery, 31(3):606–660, May 2017.

[3] A. Bagnall, J. Lines, J. Hills, and A. Bostrom. Time-series classification with cote: The collective of transformation-based ensembles. IEEE Transactions on Knowledge and Data Engineering, 27(9):2522–2535, September 2015.

[4] Aaron Bostrom and AnthonyBagnall. Binary shapelet transform for multiclass time series classification. In Sanjay Madria and Takahiro Hara, editors, Big Data Analytics and Knowledge Discovery, pages 257–269, Cham, 2015. Springer International Publishing.

[5] Jason Lines and Anthony Bagnall. Time series classification with ensembles of elastic distance measures. Data Mining and Knowledge Discovery, 29(3):565–592, May 2015.

[6] Jason Lines, Luke M. Davis, Jon Hills, and Anthony Bagnall. A shapelet transform for time series classification. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'12, pages 289–297, New York, NY, USA, 2012. ACM.

[7] Abdullah Mueen, Eamonn Keogh, and Neal Young. Logical-shapelets: An expressive primitive for time series classification. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11, pages 1154–1162, New York, NY, USA, 2011. ACM.

[8] Mustafa Gokce Baydogan, George Runger, and Eugene Tuv. A bag-of-features framework to classify time series. IEEE Trans. Pattern Anal. Mach. Intell., 35(11):2796–2802, November 2013.

[9] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax: a novel symbolic representation of time series. Data Mining and Knowledge Discovery, 15(2):107–144, October 2007.

[10] P. Senin and S. Malinchik. Sax-vsm: Interpretable time series classification using sax and vector space model. In 2013 IEEE 13th International Conference on Data Mining, pages 1175–1180, December 2013.

[11] Patrick Schäfer. The boss is concerned with time series classification in the presence of noise. Data Min. Knowl. Discov., 29(6):1505–1530, November 2015.

[12] Patrick Schäfer and Mikael Högqvist. Sfa: A symbolic fourier approximation and index for similarity search in high dimensional datasets. In Proceedings of the 15th International Conference on Extending Database Technology, EDBT'12, pages 516–527, New York, NY, USA, 2012. ACM.

[13] Z. Wang, W. Yan, and T. Oates. Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 1578–1585, May 2017.

[14] Schäfer, Patrick and Leser, Ulf. Fast and Accurate Time Series Classification with WEASEL. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM'17, pages 637–646, Singapore, Singapore, 2017. ACM.

[15] Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'14, pages 392–401, New York, NY, USA, 2014. ACM.

[16] Chen, Yanping and Keogh, Eamonn and Hu, Bing and Begum, Nurjahan and Bagnall, Anthony and Mueen, Abdullah and Batista, Gustavo. The UCR Time Series Classification Archive, $www.cs.ucr.edu/eamonn/time\_series\_data/$, July 2015.

[17] J. W. Taylor and P. E. McSharry and R. Buizza. Wind Power Density Forecasting Using Ensemble Predictions and Time Series Models. IEEE Transactions on Energy Conversion, 24(3):775–782, September 2009.

[18] Jebb, Andrew T. and Tay, Louis and Wang, Wei and Huang, Qiming. Time series analysis for psychological research: examining and forecasting change. Frontiers in Psychology, 6:727, 2015.