Forecasting without historical data

Rafiq Kamal
Centre for Applied Mathematics and Bioinformatics, Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, Kuwait and Department of English, Gulf University for Science and Technology, Kuwait

Kamaludin Dingle*
Centre for Applied Mathematics and Bioinformatics, Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, Kuwait

Boumediene Hamzi
Department of Mathematics, Imperial College London, South Kensington, UK
(Dated: February 21, 2022)

Is it possible to forecast a time series without first fitting to historical data? Here we use results derived from algorithmic information theory to make generic forecasts of series patterns in real-world time series taken from the World Bank Open Data project. We find evidence of simplicity bias — an exponential upper bound decay in probability with pattern complexity — in these series, and make *a priori* predictions regarding the probabilities of certain discretised series patterns arising in the data. Hence we perform a kind of forecasting without training data. We examine different discretisation methods, and compare them finding that not all methods yield the same results. Finally we make predictions about which of two discretised series is more likely with accuracy up ∼85%, considerably higher than a 50% baseline rate. Simplicity bias has been studied in several fields including biology, physics, and engineering models, and here we expand the empirical study and application to time series, thus potentially opening a new perspective on time series analysis.

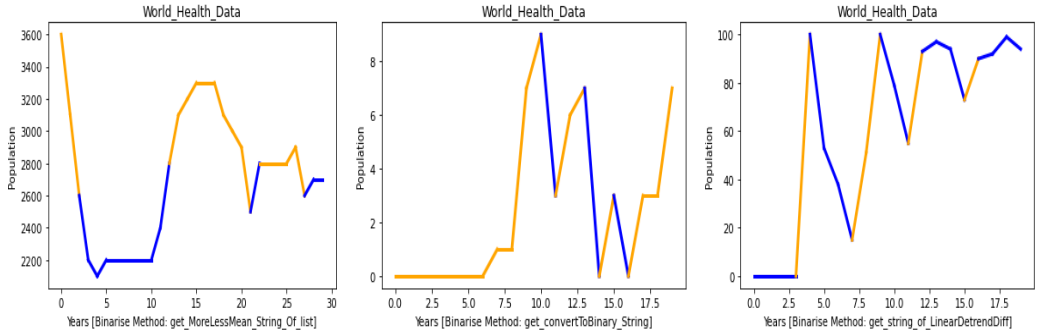
Keywords: Time series; simplicity bias; Kolmogorov complexity; algorithmic probability; forecasting

I. INTRODUCTION

Suppose you were challenged to forecast a time series, but you did not have any historical data from which to build or fit a model. Because essentially all time series predictions are made from fitting to past observations, this challenge appears impossible to meet. Is it nonetheless possible to make some generic predictions about universal patterns of behaviour which might apply in broad range of real-world contexts? This type of question has been studied (albeit in an abstract way) in a branch of computer science known as *algorithmic information theory* [1–4] (AIT). The central quantity of AIT is *Kolmogorov complexity*, $K(x)$, which measures the complexity of an individual object x via the amount of information required to describe or generate x. Within AIT, Levin's [5] coding theorem establishes a fundamental connection between $K(x)$ and probability predictions in very general settings. Loosely, the theorem states that when considering the chances that some output x (eg a binary string, a discrete pattern, a shape) is produced via some generic computation mechanism, if x is simple it is much more likely than if x is complex. More formally, $P(x) \sim 2^{-K(x)}$ where $P(x)$ is the probability that an output x is generated by a (prefix optimal) universal Turing machine fed with a random binary program input. $P(x)$ is known as the *algorithmic probability* of x, and the associated full distribution is known as the *universal distribution* [6]. Hence, under general settings, given no other information, predicting simple outputs is more likely than to be successful than more complex. This preference for simplicity is connected closely with Occam's razor, a fundamental principle of scientific reasoning that simpler explanations or models take preference over more complex ones [7].

The universal distribution and AIT are typically difficult or impossible to apply in real-world settings due to the fact that $K(x)$ is uncomputable, and the theorems are based on universal Turing machines, and the asymptotic results. However, many successful applications of AIT have been made based on various forms of approximations, for example in bioinformatics [8, 9], physics [10], signal denoising [11], among many other applications [4]. Moreover, a weaker form of the coding theorem, applicable in real-world contexts, was studied and it was shown to be very successful empirically in a range of input-output maps such as RNA sequence-to-structure map, differential equation parameter-to-solution profile map, and others [12, 13]. The authors of [12] termed this phenomenon of an inverse relation between complexity and probability *simplicity bias* (SB). Recently Johnston et al [14] have used these simplicity bias results to predict the frequency in nature of various shapes and patterns, specifically self-assembling tiles (polyominos), protein structures, natural RNA structures, and genetic network profiles. Earlier, algorithmic probability estimates were empirically studied in generic small Turing machines [15–17]. Taken together, these results suggest that algorithmic probability can be usefully applied to real-world settings to make predictions, in a wide range of settings. In this work, we apply these ideas to time se-

* Correspondence: dingle.k@gust.edu.kw



(a) (b) (c)

FIG. 1. Illustrating the different binarising methods for example time series (a) High/Low (HL) method, values above the mean value of the series are drawn in blue, depicting 1; (b) Up/Down (UD) method, where regions with a positive slope are drawn in blue assigned 1, and negative or flat drawn in yellow and assigned a 0; (c) Up/Down after applying a linear detrend (Det), similar to UD method. Update these figures with correct axis labels. Remove titles. Which time series are these? energy? Population? which country? Remove the discretisation method from the x axis, but put it in the figure name instead.

ries to test how far predictions can be made for real series, just based on generic universal probability assignment arguments, without recourse to fitting to training data.

Note that this work is not focussed on a general investigation of complexity measures of time series [18, 19], nor does it argue for one complexity measure over another [20]. Rather, we are interested in applying ideas from AIT as a mathematical framework to make non-trivial prediction, data-free predictions.

II. RESULTS

A. Theory

The Kolmogorov complexity $K_U(x)$ of a string x with respect to U, is defined [1–3] as

$$K_U(x) = \min_p \{|p| : U(p) = x\} \tag{1}$$

where p is a binary program for a prefix optimal UTM U, and $|p|$ indicates the length of the program p in bits. Due to the invariance theorem [4] for any two optimal UTMs U and V, $K_U(x) = K_V(x) + O(1)$ so that the complexity of x is independent of the machine, up to additive constants. Hence we conventionally drop the subscript U in $K_U(x)$, and speak of 'the' Kolmogorov complexity $K(x)$. Informally, $K(x)$ can be defined as the length of a shortest program that produces x, or simply as the size in bits of the compressed version of x. If x contains repeating patterns like $x = 1010101010101010$ then it is easy to compress, and hence $K(x)$ will be small. On the other hand, a randomly generated bit string of length n is highly unlikely to contain any significant patterns, and hence can only be described via specifying each bit separately without any compression, so that $K(x) \approx n$ bits. Other more expressive names for $K(x)$ are *descriptional complexity*, *algorithmic complexity*, and *program-size complexity*, each of which highlight the idea that $K(x)$ is measuring the amount of information to describe or generate x precisely and unambiguously. Note that Shannon information and Kolmogorov complexity are related [21], but differ fundamentally in that Shannon information quantifies the information or complexity of a random source, while Kolmogorov complexity quantifies the information of individual sequences or objects. More details on AIT can be found in refs. [4, 22, 23]. $K(x)$ is more technically defined as the length of a shortest program which runs on an optimal prefix *universal Turing machine* (UTM) [24], generates x, and halts.

Levin's coding theorem [5] states that

$$P(x) = 2^{-K(x)+O(1)} \tag{2}$$

where $P(x)$ is the probability that UTM U generates output string x on being fed random bits as a program (again we have dropped the subscript U). Thus, high complexity outputs have exponentially low probability, and simple outputs must have high probability. This is a profound result which links notions of data compression and probability in a direct way. $P(x)$ is also known as the *algorithmic probability* of x.

As discussed in the Introduction, applying results from AIT is not straightforward, and typically results and complexity estimate must be approximation. Conscious of these points, we recently studied coding theorem-like behaviour in real-world input-output maps, leading to our observation of a phenomenon called *simplicity bias* (SB) [12] (see also ref. [25] for commentary on that paper). SB is captured mathematically as

$$P(x) \leq 2^{-a\tilde{K}(x)-b} \tag{3}$$

where $P(x)$ is the (computable) probability of observing output x on random choice of inputs, and $\tilde{K}(x)$ is the approximate Kolmogorov complexity of the the output x: complex outputs from input-output maps have lower probabilities, and high probability outputs are simpler. The constants $a > 0$ and b can be fit with little sampling and often even predicted without recourse to sampling [12]. Examples of systems exhibiting SB are wide ranging, and include molecular shapes such as protein structures and RNA [14], finite state machines outputs [13], as well as models of financial market time series and ODE systems [12],

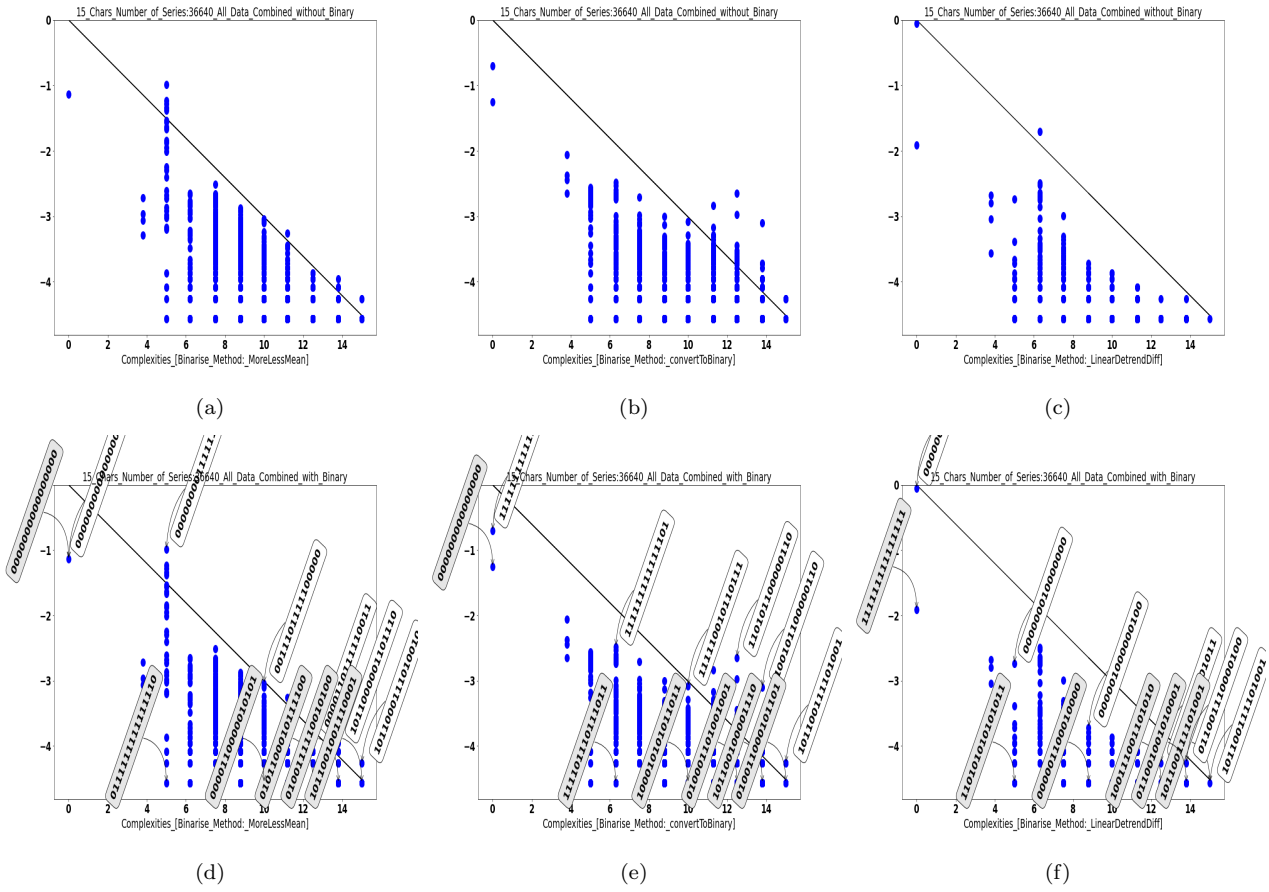


FIG. 2. Simplicity bias (SB) in time series combined data set from World Bank, using different coarse graining methods. Each blue dot is a different binary strings of length $n = 15$ bits. (a) Low/High (LH) discretisation; (b) Up/Down (UD) discretisation; and (c) Detrended up/down (Det) discrisation. (d)—(f) show the same as (a)—(c) but now with example binary strings displayed. SB is apparent in each case, with a decay in upper bound probability with increasing complexity. The upper bound (black line) prediction from Eq. (3) describes the upper bound decay int he data quite well.Add log10 P(x) to y-label, and $\tilde{K}_{HL}(x)$, $\tilde{K}_{UD}(x)$ and $\tilde{K}_{Det}(x)$ to the axes. Remove the titles. Some of the binary strings are not showing on the image, they are leaving the boundaries of the image. The numbers on the axes need to be bigger.

among others. The ways in which SB differs from Levin's coding theorem include that it does not assume UTMs, uses approximations of complexities, and for many outputs $P(x) \ll 2^{-K(x)}$. Hence the abundance of low complexity, low probability outputs [13] is a signature of SB.

A full understanding of exactly which systems will, and will not, show SB is still lacking, but the phenomenon is expected to appear in a wide class of input-output maps, under fairly general conditions. Some of these conditions were suggested in ref. [12], including (1) that the number of inputs should be much larger than the number of outputs, (2) the number of outputs should be large, and (3) that the map should be 'simple' (technically of $O(1)$ complexity) to prevent the map itself from dominating over inputs in defining output patterns. Indeed, if an arbitrarily complex map was permitted, outputs could have arbitrary complexities and probabilities, and thereby remove any connection between probability and complexity. Finally (4), because many AIT applications rely on approximations of Kolmogorov complexity via standard lossless compression algorithms [26, 27] (but see [15, 16] for a fundamentally different approach), another condition proposed is that the map should not generate pseudo-random outputs like $\pi = 3.1415...$ which standard compressors cannot handle effectively. The presence of such outputs may yield high probability outputs which appear 'complex' hence apparently violating SB, but which are in fact simple.

B. Discretising time series

To study simplicity bias (SB) in time series, we obtained data for several series from World Bank Open Data `https://data.worldbank.org/`, which gathers generic data sets of political, economic, and geographical interests, such as health, education, and energy statistics collected over many years. The choice of data set can be important when studying SB, and we expect SB to be more pronounced for 'natural' datasets, and not merely any arbitrary ones. For example, if a data set was chosen to contain particularly complex and difficult to understand series, then it is unlikely that SB would be observed, due to the

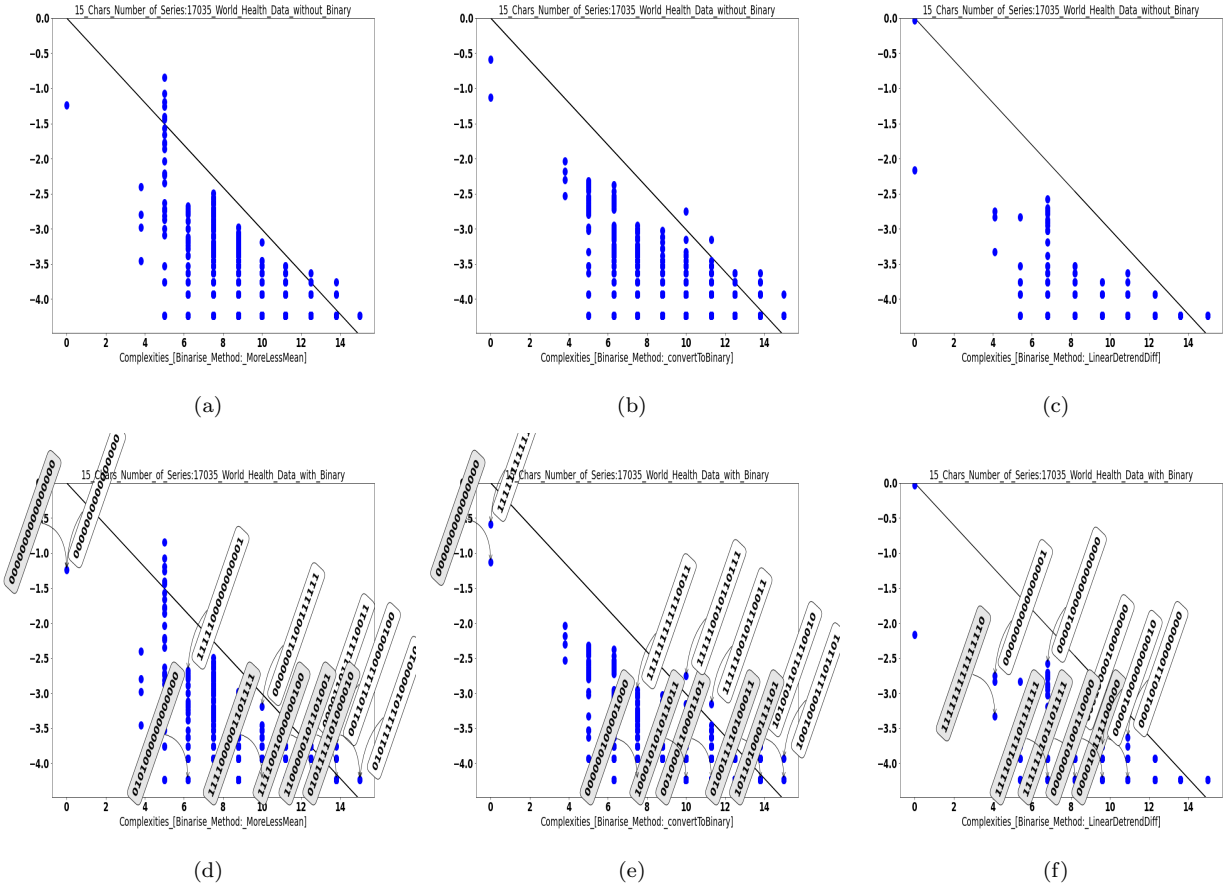


FIG. 3. Simplicity bias (SB) in time series world health data, using different coarse graining methods. Each blue dot is a different binary strings of length $n = 15$ bits. (a) Low/High (LH) discretisation; (b) Up/Down (UD) discretisation; and (c) Detrended up/down (Det) discrisation. (d)—(f) show the same as (a)—(c) but now with example binary strings displayed. SB is apparent in each case, with a decay in upper bound probability with increasing complexity. The upper bound (black line) prediction from Eq. (3) describes the upper bound decay int he data quite well.

intentional bias towards more complex series. The World Bank data then is a reasonable choice, because the time series contained were not chosen for any particular features of the series, but are a natural generic collection of data.

SB and results from AIT are most easily applied in the context of binary strings, so the real-valued time series were converted into binary strings. Let $s(t)$ be the real-valued time series. We discretise s to a binary string $x(t) \in \{0,1\}^n$ of length n. In principle, there are many different ways this can be done. One option is to calculate the mean value of the series s, and then replace $s(t)$ with 1 if $s(t) >$mean(s), and replace $s(t)$ with 0 if $s(t) \leq$mean(s). This is a kind of high/low discretisation. We will denote the complexity of the binary string x resulting from this high/low (HL) method as $\tilde{K}_{HL}(x)$. A second option to use the 'up/down' method [12, 28, 29], where if a series increases $ds/dt > 0$ from time step t to $t+1$ then the $s(t)$ is replaced by a 1, and $s(t)$ is replaced by a 0 if $dx/dt \leq 0$. We will denote the complexity of the binary string resulting from this up/down (UD) method as $\tilde{K}_{UD}(x)$. In this work, we make a slight adjustment to this UD method, to make a third option: first performing a linear detrending of the series, ie subtracting a linear fit from the series $s - (\alpha t + \beta)$, and then converting the resulting up-down series into a binary string according to the UD method. We call this detrended updown/method method *Det*. The reason for introducing this third option is that many natural time series have a monotonic increasing or decreasing trends, leading to many discretised series becoming 000...000 or 111...111. We will denote the complexity of the binary string resulting from this detrended up/down method as $\tilde{K}_{Det}(x)$.

To illustrate the discretisation process, Figure (1)(a) shows a series discretised by the HL method, and where yellow sections denote high (above mean) $s(t)$ values which are converted to 1, and blue segments for low $s(t)$ values, which are converted to 0. The corresponding binary string is XXXRAFIQ DO YOU KNOW?. Note that the segments $s(t)$ to $s(t+1)$ are coloured according to the value of $s(t)$, so it may be that, for example, some yellow segments appear to dip into the lower half, but this is merely due to the fact that $x(t)$ is high while $s(t+1)$ is low. Figure (1)(b) illustrates the UD method, where the sign of the slope ds/dt is used to define the binary string. The corresponding binary string is XXXRAFIQ DO YOU KNOW? Figure (1)(c) shows the analogous plot for the for the linear detrended up/down method. Here yellow corresponds to increasing

segments (after detrending) and blue decreasing. Finally Figure (1)(c) shows the *Det* method, which is fundamentally quite similar to the UD method. The corresponding binary string is XXXRAFIQ DO YOU KNOW?

Note that, as with any coarse graining or discretisation, many different series will correspond to a given binary pattern. Hence we are not predicting a particular real-values series, but rather predicting the probability of a given discretised series pattern.

C. Pattern probability bounds from complexities

Empirically, SB has been observed in systems with small output patterns, eg of $n \approx 50$ bits [12] or smaller [15, 16] in some cases. However, both in theory and in practice SB is more likely to be observed for larger systems, where the size of the output pattern/string n is larger. With sufficient data, a larger value of n should lead to more clear SB. With limited data, because there are up to 2^n different binary series this means that large values of n will lead to very small frequency counts per strings, leading to large errors in estimates of $P(x)$, resulting is less clear SB. Given our data sets which have about in the order of 10^3 to 10^4 series, we choose $n = 15$ bits (but see the Appendix for other sizes n).

Two main predictions are to be tested with the data: firstly, that an inverse relation of complexity and probability is observed ie SB; and secondly that the upper bound on the data follows that of Eq. (3). Due to various factors regarding the how upper bound is predicted, the slope a or intercept b may not be accurately predicted, even though the SB phenomenon is observed.

We begin with probability predictions for a combined data set amalgamated from various areas namely health, education, energy, and economics. This combined data set contains ∼36,000 series. Figure 2(a) results from using the HL method. SB is apparent in this plot, and the upper bound prediction (black line) describes the upper bound quite closely. Note that the only information used to make this predictions is the assuming that most of the $2^n = 2^{15}$ different possible binary strings are in principle realisable (Rafiq, or are we using the maximum number of series found? I can't remember.), and making the default assumption that $b = 0$. There is no fitting to the data, except to know the mean series value which is used to specify whether a value is high (H) or low (L). In a scenario where a mean value can be known or estimated *a priori* then even this value need not be fit. For example, it might be known from physical constraints that $s(t)$ must be bounded by some values $s(t) \in [p, q]$ and perhaps have mean $\sim (p + q)/2$.

Results from applying the UD method are given in Figure 2(b), and again SB is observed but it is somewhat noisier, especially around $\tilde{K}_{UD}(x) \approx$13 bits of complexity where the is a spike of overabundance. The upper bound prediction line describes the true data value decay less closely. Figure 2(c) shows a plot of the same data but for *Det* method, in which SB is quite clear, with a clear relation decay in probability with increasing complexity. The upper bound describes the data quite closely, but not as well as for the HL method. Figures 2(d)—(f) show the same plots as Figure 2(a)—(c) but now with some binary strings displayed. Specifically, examples of the highest and lowest probability strings for each complexity value as plotted.

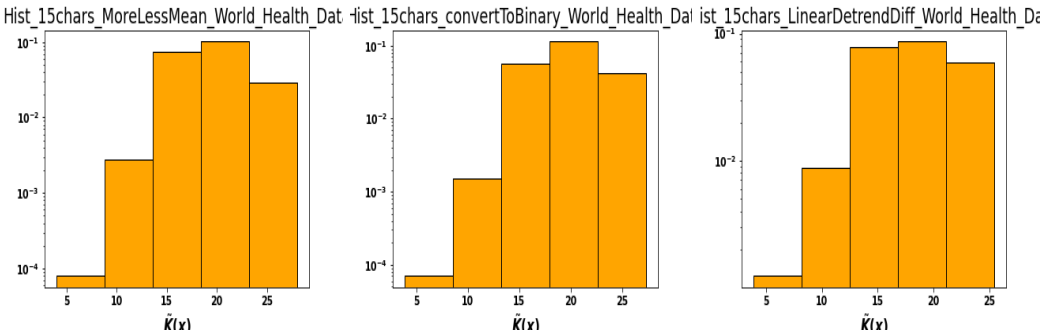
As reported earlier for different systems [12], many output binary string patterns also fall far below the upper bound black line. Indeed, most patterns fall quite far below the bound, but at the same time most of the probability mass is accounted for by points which are close to the bound [12]. Thus, for a randomly selected series, it corresponding binarey string pattern is likely to be close to the bound with high probability [13]. In ref. [13] this phenomenon of low-complexity, low-probability outputs was studied further, and it was suggested that such patterns are those that are intrinsically not very complex, but are difficult to make for particular system. Nonetheless, this phenomenon is still not fully understood, and in a sense these points far below the bound represent a failing of our probability predictions. Hence a better grasp of why such patterns arise and which patterns have this property would help in improving probability predictions.

In Figure 3 we study SB in world health time series, this time focussing in on one particular sector— health — to see if SB persists for particular fields or perhaps only on average over different fields. The health sector was chosen for demonstration because it has a large number of series (∼17,000), whereas many other fields have much less data, for example nitrous oxide emissions for which only a few hundred series could be found. The general trends of the graphs are similar to the combined data set of Figure2, with clear SB observed in all three methods. In particular, the UD method shows more pronounced SB as compared to in the combined data, and the bound is followed quite closely, but the slope is not quite as accurate. Additionally, the upper bound prediction agrees with the data bound quite well. The *Det* method is not as compelling XXX.

III. COMPLEXITY HISTOGRAMS

The distribution of complexity values obtained depends both on the probability of each series pattern, and the number of patterns with each complexity. Hence even if the SB bound of Eq. (3) describes the upper bound well, it does not imply that a random series will be simple, or that the average complexity of random series will be simple. In general Eq. (3) describes an exponential decay in probability with complexity, and it is known that the number of patterns increases exponentially with complexity [4], but these two exponential trends may not be perfectly balanced, and hence may tip the distribution of complexities either to higher or lower values.

In Figure 4 we show the distribution of complexity values for each method, for the world health data. As can be seen, there is a modest leaning towards higher complexity values. The mean complexity for each is XX, as compared to the mean for purely random strings which is XX with standard deviation XX. So even though they lean to more complex, they are still notably simpler than random strings.



(a) (b) (c)

FIG. 4. The distribution complexity value for the world health data, for (a) HL, (b) UD, and (c) Det methods. Simplicity bias (SB) by itself does not guarantee that only simple outputs will arise with higher probability. In all three methods, there is a tendency towards higher complexity values what is the mean complexity for each historgram? compare to mean for random strings

IV. PREDICTING WHICH OF $P(x)$ OR $P(y)$ IS HIGHER

It is often interesting to know be able to predict whether $P(x) > P(y)$ or if instead $P(x) < P(y)$. For example in investing, one might want to know which outcome to bet on, x or y. In this prediction, both the constants a and b are irrelevant, because choosing which of $2^{-a\tilde{K}(x)-b}$ and $2^{-a\tilde{K}(y)-b}$ is largest does not depend on these constants. Indeed, the prediction just depends on the relative size of $\tilde{K}(x)$ and $\tilde{K}(y)$. So even if we could not estimate a or b, this ratio prediction method is still valid for determining which outcome x or y is more likely.

For our World Bank data sets, we tested this prediction method, while also accommodating the situation where $\tilde{K}(x) = \tilde{K}(y)$. In more detail, for a given data set, we chose two real valued series at random from the dataset, then converted them both to binary strings according to the HL method. Denoting the resulting binary strings with x and y, we then predict

$$P(y) > P(x) \quad \texttt{if} \quad \tilde{K}(y) < \tilde{K}(x) \tag{4}$$
$$P(y) < P(x) \quad \texttt{if} \quad \tilde{K}(y) > \tilde{K}(x) \tag{5}$$
$$P(y) > P(x) \ \texttt{w/prob. 0.5 if} \ \tilde{K}(y) = \tilde{K}(x) \tag{6}$$

For the last condition just stated, a random number is drawn to guess whether $P(y) > P(x)$ or $P(y) < P(x)$ each with 50% probability. It may seem more natural to predict that $P(y) = P(x)$ when $\tilde{K}(y) = \tilde{K}(x)$, but following this prediction will almost certainly lead to erroneous predictions most of the time. The reason for this is that the probabilities are measured to much finer grained levels of accuracy, so that it is highly unlikely that $P(y) = P(x)$, whereas the complexities are far more coarsely measured, and hence coincidences of complexities values are much more likely. According this prediction protocol just described, if complexity had no role in modulating probabilities, then the success rate should be 50% in theory. Testing out this prediction protocol with the same health data studied above we have the following success results:

Combined Data, HL method, success rate 81%
Combined Data, UD method, success rate 84%
Combined Data, Det method, success rate 59%

World Health Data, HL method, success rate 75%
World Health Data, UD method, success rate 88%
World Health Data, Det method, success rate 55%

XX Rafiq - how many samples did you take to find these average percentages?

So we see that without fitting to the data, and simply using the relative size of the complexity as a predictor, success rates well above random null model can be achieved. Of the three methods, The UD method had the highest success rate followed closely by the HL method. The *Det* performed quite badly, only marginally above the 50% prediction. This can be understood by noting that in Figure 2(c) and Figure 3(c) the vast majority of the probability mass is taken up by only 2 or 3 data points, for which there is little local relation to complexity. Hence, even though SB is observed over orders of magnitude of probability and a large range of complexity values of $\tilde{K}_{Det}(x)$, the trend does not hold for the very highest probability patterns and hence the predictions are less impressive. Nonetheless, the fact that we can predict which series is more likely with the HL and UD methods quite accurately, while utilising only minimal details of the map, is quite surprising and promising. Indeed, the UD method uses essentially no details at all, it is just based on forecasting certain up/down patterns.

V. COMPARISON TO BROWNIAN MOTION

Brownian motion is used in time series analysis as a simple baseline model of series evolution over time. Given that we have observed SB in real time series, it is interesting to compare to artificial series from Brownian motion also. Indeed, in the Supplementary Information of [12] SB was briefly studied both numerically and analytically in the

case of random walks with a HL method, and evidence of SB was found.

We numerically generated 10,000 Brownian motion series via the cumulative summation of Gaussian random variables. Figure 5 shows the resulting probability-complexity plots. Starting with the HL method in panel (a), there is certainly some SB observed, but the bound is not followed closely, and also the slope of the decay in probability is less steep. Indeed, the difference in $\log_{10}$ probability between the highest and lowest patterns appears less stark. Turning to panel Figure 5(b), the SB has disappeared, hence this marks a strong contrast between the Brownian motion and both the combined dataset and health dataset. This lack of SB is expected in fact, because in Brownian motion increasing and decreasing at each time step is a Markov process independent of past values, and hence the UD profile is essentially a random string of 0s and 1s. Therefore there is no bias, and no SB. Finally in Figure 5(c), the *Det* method shows quite clear SB, somewhat in contrast to the rea-world combined and health datasets for which *Det* showed less clear SB.

VI. DISCUSSION

We have numerically studied simplicity bias (SB) and forecasting in discretised real-world time series datasets. We studied different discretisation procedures also. Our main finding is that we report observing SB, and that we can make non-trivial predictions regarding the probability of different time series patterns, without using almost any details or fitting to the historical data. In this manner, we present a kind of forecasting without historical data. Of course we are not claiming that accurate quantitative predictions can be made for series without fitting to trends in historical data, but we are claiming that even without recourse to such training data non-trivial forecasts about patterns of trajectories can be made. We also showed that predictions about which of two randomly chosen series has higher probability can be done with high success rate, up to 85%. With our data sets, these success rates were high for the HL and UD methods, but not the Det method.

A weakness of our predictions is that they only constitute an upper bound on the probabilities, many output trajectory patterns x fall far below their respective upper bounds. Following the hypothesis from [13], these low-complexity low-probability outputs are presumably patterns which the logistic map finds 'hard' to make, yet are not intrinsically very complex. Further, the presence of these low-complexity low-probability patterns may indicate the non-universal power of the map [13].

Copied from other paper, cut The application of AIT to real-world science problems suffers from several problems, including that (a) Kolmogorov complexity is uncomputable, (b) the results are framed and proved in the context of universal Turing machine (UTMs) while many real world maps are not Turing complete, and (c) results are valid up to to $O(1)$ terms and therefore, strictly, only accurate in the asymptotic limit of large complexities. Given these, it is surprising that AIT and Levin's coding theorem can be successfully applied at all. However, several lines of reasoning can help us understand why they are in fact applicable, at least approximately. Firstly in response to (a), although technically uncomputable, Kolmogorov complexity is fundamentally merely a measure of the size in bits of the compressed version of a data object. Hence, Vitanyi [11, 30] points out that because naturally generated data is unlikely to contain pseudo random complexities like π, the true complexity is unlikely to be much shorter than that achievable by every-day compressors. For more on this discussion, see ref. [30] and ref. [31] for work on short program estimates via short lists of candidates with short programs. Secondly, in response to (b) it is worth noting that the SB bound is specifically relevant in the computable (ie non-UTM) setting. Moreover building on pioneering work with small Turing machines [15, 16], Zenil et al. [17] have numerically studied coding theorem behaviour at different levels of the Chomsky hierarchy and found that it persists. Indeed, they also found close agreement between complexity estimates obtained from different levels, for the short binary strings which they studied. From the theoretical side, Calude et al. [32] developed an AIT for finite state machines (the lowest on the Chomsky hierarchy), deriving analogous results, suggesting that many fundamental ideas and results from AIT need not only apply to UTMs. Moreover, given that UTMs (at the top of the Chomsky hierarchy) and finite state machines (at the bottom of the Chomsky hierarchy) share many similar mathematical results, it is not unreasonable to assume that the results hold for other systems (such as RNA folding rules) which sit between the two extremes of the hierarchy. Thirdly, we argued earlier [12] that it is common in physics that mathematical formulae apply quite accurately well outside the (eg asymptotic) regions for which they have been proven. Although it is not possible to completely remove $O(1)$ terms in AIT [33], it is still an open and interesting question in theoretical computer science why asymptotic analysis (such as ignoring $O(1)$ terms) is valid and works so well in practical applications [34].

Further reasons to understand why the AIT coding theorem should work in real-world applications can be found in information theory research developed largely independently of Levin's work. The fundamental connection between probability and data compression has also been studied by Cover [35], Langdon [36] and Rissansen [37]. Since then, different communities — eg information theory [38, 39], optimal gambling strategies [40], and password guessing [41] — have studied and exploited the probability-compression connection without utilising Kolmogorov complexity or UTMs. In a review, Merhav and Feder [42] surveyed results in the area known as *universal prediction* and explicitly point to $2^{-LZ(x)}$ as an effective universal probability assignment for prediction based on the results of ref. [43] and others, where $LZ(x)$ is the Lempel-Ziv compression complexity measure, essentially the same as we use here. Additionally, Merhav [41] uses a conditional coding theorem predictor $2^{-LZ(y|x)}$ which is again very closely analogous to the AIT conditional coding theorem relation, $2^{-K(y|x)}$ [44]. These results, together with the arguments given above in response to (a)—(c) all support and motivate the application of AIT coding theorem work in science and engineering, and also extending fundamental research related to the coding theorem [45].

The methods we describe are general and can apply to other prediction systems. In future work it would be interesting to combine universal prediction methods like SB with other machine learning methods to predict chaotic time series (eg [46]), potentially aiding or improving the predictions (see also early [47, 48] and recent [49, 50] studies on Kolmogorov complexity connections to machine learning). One direction may be to use the SB bound to guide choosing a prior, as Hutter has described [51, 52].

Acknowledgements: We thank Hamed Bajgiran (Caltech) for useful discussions. KD acknowledges financial support from a Faculty Seed Grant (number 253571) awarded by the Gulf University for Science and Technology. Boumedien? any funding to ack?

[1] R. J. Solomonoff. A preliminary report on a general theory of inductive inference (revision of report v-131). *Contract AF*, 49(639):376, 1960.

[2] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of information transmission*, 1(1):1–7, 1965.

[3] Gregory J Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM (JACM)*, 22(3):329–340, 1975.

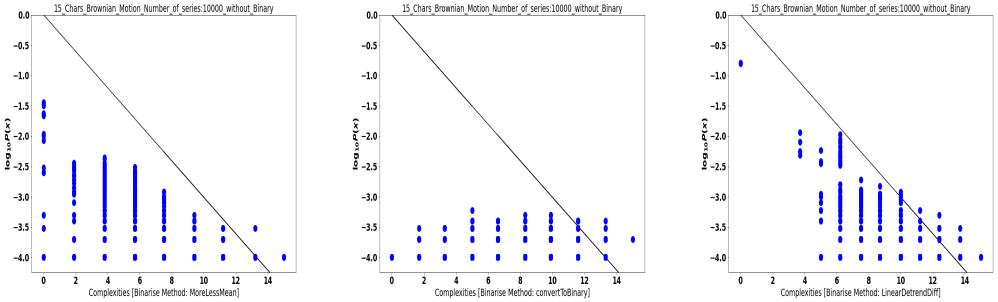
[4] M. Li and P.M.B. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag New York Inc, 2008.

[5] L.A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.

[6] Ray J Solomonoff. The kolmogorov lecture the universal distribution and machine learning. *The Computer Journal*, 46(6):598–601, 2003.

[7] Alan Baker. Simplicity. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2016 edition, 2016.

[8] R. Cilibrasi and P.M.B. Vitányi. Clustering by compression.



(a) (b) (c)

FIG. 5. Examining simplicity bias (SB) in Brownian motion. (a) the HL method shows some SB, but it is not as clear as for the real-world data; (b) the UD method shows no SB at all; (c) the *Det* method shows clear SB.

Information Theory, IEEE Transactions on, 51(4):1523–1545, 2005.

[9] P. Ferragina, R. Giancarlo, V. Greco, G. Manzini, and G. Valiente. Compression-based classification of biological sequences and structures via the universal similarity metric: experimental assessment. *BMC bioinformatics*, 8(1):252, 2007.

[10] Ram Avinery, Micha Kornreich, and Roy Beck. Universal and accessible entropy estimation using a compression algorithm. *Physical review letters*, 123(17):178102, 2019.

[11] Paul MB Vitányi. Similarity and denoising. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20120091, 2013.

[12] Kamaludin Dingle, Chico Q Camargo, and Ard A Louis. Input–output maps are strongly biased towards simple outputs. *Nature communications*, 9(1):761, 2018.

[13] Kamaludin Dingle, Guillermo Valle Pérez, and Ard A Louis. Generic predictions of output probability based on complexities of inputs and outputs. *Scientific reports*, 10(1):1–9, 2020.

[14] Iain G Johnston, Kamaludin Dingle, Sam F Greenbury, Chico Q Camargo, Jonathan PK Doye, Sebastian E Ahnert, and Ard A Louis. Symmetry and simplicity spontaneously emerge from the algorithmic nature of evolution. *Proceedings of the National Academy of Sciences*, 2022.

[15] J.P. Delahaye and H. Zenil. Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of algorithmic randomness. *Appl. Math. Comput.*, 219:63–77, 2012.

[16] Fernando Soler-Toscano, Hector Zenil, Jean-Paul Delahaye, and Nicolas Gauvrit. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PloS one*, 9(5):e96223, 2014.

[17] Hector Zenil, Liliana Badillo, Santiago Hernández-Orozco, and Francisco Hernández-Quiroz. Coding-theorem like behaviour and emergence of the universal distribution from resource-bounded algorithmic probability. *International Journal of Parallel, Emergent and Distributed Systems*, 34(2):161–180, 2019.

[18] Ling Tang, Huiling Lv, Fengmei Yang, and Lean Yu. Complexity testing techniques for time series data: A comprehensive literature review. *Chaos, Solitons & Fractals*, 81:117–135, 2015.

[19] ME Torres and LG Gamero. Relative complexity changes in time series using information measures. *Physica A: Statistical Mechanics and its Applications*, 286(3-4):457–473, 2000.

[20] William Bialek, Ilya Nemenman, and Naftali Tishby. Complexity through nonextensivity. *Physica A: Statistical Mechanics and its Applications*, 302(1-4):89–99, 2001.

[21] Peter Grunwald and Paul Vitányi. Shannon information and Kolmogorov complexity. *arXiv preprint cs/0410002*, 2004.

[22] C.S. Calude. *Information and randomness: An algorithmic perspective*. Springer, 2002.

[23] P. Gács. *Lecture notes on descriptional complexity and randomness*. Boston University, Graduate School of Arts and Sciences, Computer Science Department, 1988.

[24] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.

[25] Mark Buchanan. A natural bias for simplicity. *Nature Physics*, 14:1154, 2018.

[26] A. Lempel and J. Ziv. On the complexity of finite sequences. *Information Theory, IEEE Transactions on*, 22(1):75–81, 1976.

[27] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.

[28] Mark H Hansen and Bin Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

[29] K. Willbrand, F. Radvanyi, J.P. Nadal, J.P. Thiery, and T.M.A. Fink. Identifying genes from up–down properties of microarray expression series. *Bioinformatics*, 21(20):3859–3864, 2005.

[30] Paul Vitányi. How incomputable is kolmogorov complexity? *Entropy*, 22(4):408, 2020.

[31] Jason Teutsch and Marius Zimand. A brief on short descriptions. *SIGACT News*, 47(1):42–67, March 2016.

[32] Cristian S Calude, Kai Salomaa, and Tania K Roblot. Finite state complexity. *Theoretical Computer Science*, 412(41):5668–5677, 2011.

[33] Markus Müller. Stationary algorithmic probability. *Theoretical Computer Science*, 411(1):113–130, 2010.

[34] Scott Aaronson. Why philosophers should care about computational complexity. *Computability: Turing, Gödel, Church, and Beyond*, 261:327, 2013.

[35] TM Cover. Universal gambling schemes and the complexity measures of kolmogorov and chaitin. rep. no. 12, statistics dep, 1974.

[36] GLEN Langdon. A note on the ziv-lempel model for compressing individual sequences (corresp.). *IEEE Transactions on Information Theory*, 29(2):284–287, 1983.

[37] Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4):629–636, 1984.
[38] Meir Feder, Neri Merhav, and Michael Gutman. Universal prediction of individual sequences. *IEEE transactions on Information Theory*, 38(4):1258–1270, 1992.
[39] TM Cover and J.A. Thomas. *Elements of information theory*. John Wiley and Sons, 2006.
[40] Meir Feder. Gambling using a finite state machine. *IEEE Transactions on Information Theory*, 37(5):1459–1465, 1991.
[41] Neri Merhav and Asaf Cohen. Universal randomized guessing with application to asynchronous decentralized brute–force attacks. *IEEE Transactions on Information Theory*, 66(1):114–129, 2019.
[42] Neri Merhav and Meir Feder. Universal prediction. *IEEE Transactions on Information Theory*, 44(6):2124–2147, 1998.
[43] Eli Plotnik, Marcelo J Weinberger, and Jacob Ziv. Upper bounds on the probability of sequences emitted by finite-state sources and on the redundancy of the lempel-ziv algorithm. *IEEE transactions on information theory*, 38(1):66–72, 1992.
[44] Paul MB Vitányi. Conditional kolmogorov complexity and universal probability. *Theoretical Computer Science*, 501:93–100, 2013.
[45] Samuel Epstein. An extended coding theorem with application to quantum complexities. *Information and Computation*, 275:104660, 2020.
[46] Boumediene Hamzi and Houman Owhadi. Learning dynamical systems from data: A simple cross-validation perspective, part i: Parametric kernel flows. *Physica D: Nonlinear Phenomena*, 421:132817, 2021.
[47] Jürgen Schmidhuber. Discovering solutions with low kolmogorov complexity and high generalization capability. In *Machine Learning Proceedings 1995*, pages 488–496. Elsevier, 1995.
[48] Jürgen Schmidhuber, Jieyu Zhao, and Marco Wiering. Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement. *Machine Learning*, 28(1):105–130, 1997.
[49] Guillermo Valle-Pérez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.
[50] Santiago Hernández-Orozco, Hector Zenil, Jürgen Riedel, Adam Uccello, Narsis A Kiani, and Jesper Tegnér. Algorithmic probability-guided machine learning on non-differentiable spaces. *Frontiers in artificial intelligence*, 3, 2020.
[51] Marcus Hutter. On universal prediction and bayesian confirmation. *Theoretical Computer Science*, 384(1):33–48, 2007.
[52] Marcus Hutter. *Universal artificial intelligence: Sequential decisions based on algorithmic probability*. Springer Science & Business Media, 2004.

Appendix A: Methods

1. Time series data

World Bank Open Data `https://data.worldbank.org/` Looked for complete lines of series, most recent
Xwrite wiriteX

2. Complexity estimation

To estimate complexity in ref. [12] we used

$$C_{LZ}(x) = \begin{cases} \log_2(n), & x = 0^n \text{ or } 1^n \\ \log_2(n)[N_w(x_1...x_n) + N_w(x_n...x_1)]/2, & \text{otherwise} \end{cases} \tag{A1}$$

where the number of words $N_w(x)$ forms the basis for this complexity measure Lempel and Ziv [26], and where the simplest strings 0^n and 1^n are separated out because $N_w(x)$ assigns complexity $K = 1$ to the string 0 or 1, but complexity 2 to 0^n or 1^n for $n \geq 2$, whereas the true Kolmogorov complexity of such a trivial string actually scales as $\log_2(n)$ for typical n, because one only needs to encode n. Having said that, the minimum possible value is $K(x)\approx 0$ for a simple set, and so eg for binary strings of length n we can expect $0 \lesssim K(x) \lesssim n$ bits. Because for a random string of length n the value $C_{LZ}(x)$ is often much larger than n, especially for short strings, we scale the complexity so that a in Eq. (3) is set to $a = 1$ via

$$\tilde{K}(x) = \log_2(M) \cdot \frac{C_{LZ}(x) - \min_x(C_{LZ})}{\max_x(C_{LZ}) - \min_x(C_{LZ})} \tag{A2}$$

where M is the maximum possible number of output patterns in the system, and the min and max complexities are over all strings x which the map can generate. $\tilde{K}(x)$ is the approximation to Kolmogorov complexity that we use throughout. This scaling results in $0 \leq \tilde{K}(x) \leq n$ which is the desirable range of values. We will assume that $b = 0$ throughout, which is a default assumption as argued and discussed in [12].

3. Rafiq - average complexity

XXXRafiq: please can you run this code

```
all_bin_strings_len_15=[np.binary_repr(i,15) for  i in range(2**15)]
```

and compute the (scaled) complexity for each string. Then find the mean of the complexity values in the list, and the standard deviation of the complexity values in the list.

you can use **np.mean()** and **np.std()** for the standard deviation. Thanks

Appendix B: Other examples

XX Nitrous oxide too little data. Many have too little data < 1000 samples.

In this section we illustrate the resulting pattens when for a given dataset, different values of series length n are used. Additionally, as another sector example, here we use world development indicators. Figure 6 shows complexity-probability plots for $n = 5, 10, 15$ and 20 bit series. There are about 8,700 series in this dat set. We focus on just the HL method. With very small $n = 5$, the plot is noisy with little clear relation between $P(x)$ and $\tilde{K}_{HL}(x)$. With $n = 10$ and $n = 15$, clear SB is observed in this dataset. Moving up to $n = 20$, there are now exponentially more possible patterns, and so there are too few data in this set to share between them, leading to large errors in frequency. The SB is less clear, especially in the tail at higher complexity values. Many of those high complexity patterns have over exagerated frequencies due to partial sampling. Nonetheless, SB is still present in this dataset.

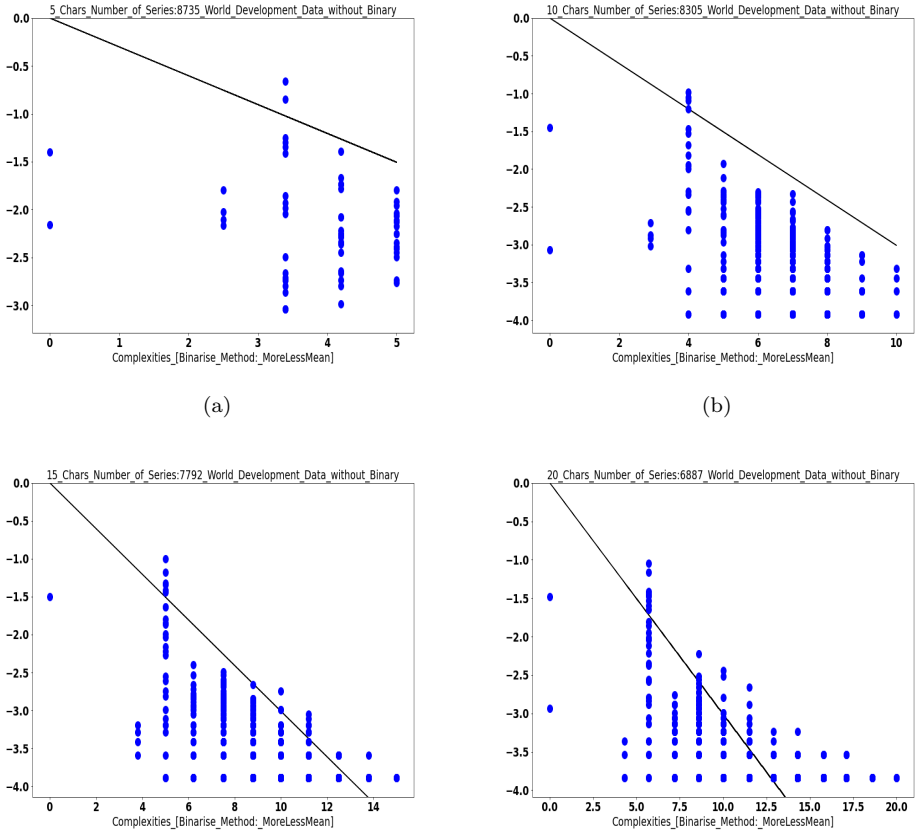


FIG. 6. World development indicators data with HL method, for different lengths (a) $n = 5$, (b) $n = 10$, (c) $n = 15$, (d) $n = 20$