Time Series Smoothing and Downsampling for Improving Forecasting Accuracy

Vadim Romanuke*

Polish Naval Academy, Gdynia, Poland

Abstract – Both statistical and neural network methods may fail in forecasting time series even operating on a great amount of data. It is an open question of which amount fits best to make sufficiently accurate forecasts on it. This implies that the length or density of time series might be optimised. Hence, the objective is to try to improve the quality of forecasting by an assumption that model parameters are set nearly at their optimal values. To achieve the objective, the two types of the benchmark time series are considered: sine-shaped series and random-like series with repeatability. Trend, seasonality, and decay properties are embedded into each type. Based on the benchmark of 24 time series models, it is ascertained that, for improving the quality of forecasting, the time series should be smoothed and then downsampled. These operations can be fulfilled successively until the improvement fails. If preliminary smoothing worsens forecasts, the raw time series is straightforwardly downsampled until the forecasting accuracy starts dropping. However, if the time series has a visible property of being noised, the preliminary smoothing is strongly recommended.

Keywords – Autoregressive integrated moving average (ARIMA), downsampling, forecasting accuracy, long short-term memory (LSTM), maximum absolute error (MaxAE), root-mean-square error (RMSE), smoothing, time series forecasting.

I. INTRODUCTION

Time series forecasting is a fascinating field of methods to control and predict processes comprising sequences of data. A common example of a data sequence to be forecast is a set of successive equally spaced points in time [1], [2], which have a definite impact and thus are studied. The methods of time series forecasting found a lot of practical applications in industrial, socio-economic, ecological, entertainment, and scientific areas [1], [3], [4]. In particular, time series forecasting is extensively used in control engineering, communication, financial marketing, social networking, meteorology, seismology, geology, geophysics, signal detection and estimation, clustering, classification, anomaly detection, segmentation, optimization, etc. Recently, due to bad growth of SARS-CoV-2 all over the world, time series forecasting has become an important way for medics and governments to prevent catastrophic aftermaths of the disease.

In general, a time series forecasting problem can be tried to get solved by statistical forecasting methods considering averages, regression factors, and scedasticity of data [2], [3].

However, none of these methods is universally suitable for time series. Thus, the family of autoregressive-moving-average (ARMA) models is better for smoother series [3], whereas an ARMA-based model may not follow a time series with outliers [2], [5]. On the contrary, the family of the generalized autoregressive conditional heteroscedasticity (GARCH) models works with stochastic volatility better than ARMA models [3], but a GARCH-based model is worse at smoother series [5], [6]. Moreover, both the ARMA and GARCH approaches may be too inaccurate for time series having heavy trend and seasonal components [2], [3], [5], [7]. Furthermore, although some modifications of these approaches like autoregressive integrated moving average (ARIMA) and seasonal ARIMA (SARIMA) can handle seasonality and trend effects, they require adjustment of a set of additional parameters [3], [5]. For instance, the ARIMA approach requires a degree of the non-seasonal differencing polynomial and an innovation distribution specification. In its turn, the SARIMA approach requires specification of seasonal autoregressive and moving-average polynomial coefficients [8]. Obviously, these adjustments take much time themselves as they are fulfilled commonly by researchers (users) rather than machines, so an eventual forecasting may become just needless. Therefore, time series forecasting still needs more automation in order to improve the quality of forecasting and make it independent of additional model adjustments.

II. MOTIVATION AND OBJECTIVE

According to recent papers, reports, and blogs, one of the most promising ways to simplify time series forecasting and make it more reliable (stable) is the neural network approach. Specifically, the approach consists in using either dynamic neural networks (DNNs) or deep learning networks (DLNs) [9], [10]. A standard DNN for time series forecasting is a two-layer feedforward network (2FFN), with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. The output of the 2FFN may be fed back to its input (through delays). Such a feedback loop can be opened for efficient training [9], [11].

Nevertheless, DNNs are very unstable at forecasting real practice time series (e.g., stock prices and similar highly volatile values). Moreover, apart from specifying the number of

* Corresponding author's e-mail: romanukevadimv@gmail.com

hidden neurons and the number of delays for the 2FFN architecture, a specification of the time series problem is also required. This requirement does not exist when DLNs are applied to time series forecasting. Indeed, recurrent neural networks (RNNs) based on long short-term memory (LSTM) are a promising type of DLNs for time series forecasting [10]. LSTM-based RNNs (LSTMNs) are believed to learn by any time series data, but they require instead another set-up. The set-up includes hyperparameters of an LSTMN and parameters of its training. Surely, the set-up requires its own adjustment, but the main problem may be in time series data. Thus, short and non-periodic time series cannot be tracked so that future time steps would be reliably forecast. Such data are even better processed by ARIMA-based approaches.

When the amount of data is sufficient, LSTMNs are claimed to outperform any other approach in time series forecasting [10], [12]. However, this is yet to be ascertained by a neutral and competitive analysis. The claimed capabilities of LSTMNs are not taken from scratch. First, an appropriate set-up of an LSTMN (as well as of any other DLN) involves fine-tuning a long list of very sensitive parameters: the number of LSTM layers and the number of neurons in each layer, the decision on inserting ReLU layers, dropout layers and their probabilities, the initial learning rate, the learning rate drop factor and drop period. On the other hand, a time series can be forecast by ARIMA with providing just a degree of the non-seasonal autoregressive polynomial (NSARP) and setting the degree of non-seasonal integration (NSI) at 1 [3], [5], [8]. Second, if a time series has a truly random nature within a given interval (without any repeating sequences masked by noise), neither LSTMN nor ARIMA can help.

TABLE I
TYPES OF BENCHMARK TIME SERIES AND TAGS TO THEIR PROPERTIES

Property embedded into the series	Tag to the sine-shaped series, #	Tag to the random-like series, #
None	1	2
Linear trend	3	4
Seasonality	5	6
Linear trend and seasonality	7	8
Exponential decay	9	10
Exponential growth	11	12
Linear trend with exponential decay	13	14
Seasonality with exponential decay	15	16
Linear trend and seasonality with exponential decay	17	18
Linear trend with exponential growth	19	20
Seasonality with exponential growth	21	22
Linear trend and seasonality with exponential growth	23	24

It is easy to show that both LSTMN and ARIMA-based models may fail in forecasting even operating on a great amount of data. However, it has been not yet shown which amount fits best to make forecasts on it. Strictly speaking, the length

(density) of time series might be optimised. At least, it is still an open question. Therefore, the objective is to try to improve the quality of forecasting by an assumption that model parameters are set nearly at their optimal values. To achieve the objective, the two types of the benchmark time series are to be considered: sine-shaped series and random-like series with repeatability. A trend, seasonality, and decay properties will be embedded into each type (see Table I). The criteria of comparing the forecasting results will be the commonly accepted root-mean-square error (RMSE) [1], [5] and another one presumed to give non-averaging inferences. The benchmark is to be analysed based on the criteria, without referring to operation speed or time spent on a model identification (training an LSTMN and obtaining an ARIMA model). The analysis will be discussed followed by appropriate recommendations and conclusion.

III. BENCHMARK TIME SERIES AND CRITERIA

The benchmark time series are used to compare performance of various approaches to time series forecasting. Every benchmark time series is generated by using pseudorandom numbers drawn from the standard normal distribution (with zero mean and unit variance) [7], [13], [14]. If the time series is defined on a sequence of time points $t = \overline{1, T}$, where T is an available amount of data, then a vector $\Theta(T)$ of T such pseudorandom numbers is used to simulate noise and volatility. Every next recall of vector $\Theta(T)$ produces a new pseudorandom sequence, so these sequences will be further indexed to distinguish them.

Therefore, the simple sine-shaped series is generated as a discrete function:

$$y_1(t) = [a_1 + 0.25\Theta_1(T)] \sin(\omega t) + a_2\Theta_2(T), \quad (1)$$

where $a_1 > 0$, $a_2 > 0$, and factor $\omega > 0$ indicates an oscillation frequency (or periodicity). Here and further below the index of the series (named as variable y) is its tag in Table I. The simple random-like series is generated similarly:

$$y_2(t) = [a_1 + 0.25\Theta_3(T)]r(t) + a_2\Theta_4(T), \quad (2)$$

where $r(t)$ is a sequence of identical randomly-structured subsequences (IRSS). Its periodicity P_r is defined by the subsequence.

The time series with a linear trend is obtained by just adding a product of trend strength a_3 and time t to (1) and (2):

$$y_3(t) = [a_1 + 0.25\Theta_5(T)] \sin(\omega t) + a_2\Theta_6(T) + a_3t, \quad (3)$$

$$y_4(t) = [a_1 + 0.25\Theta_7(T)]r(t) + a_2\Theta_8(T) + a_3t. \quad (4)$$

It is worth noting that vectors $\Theta(T)$ in (3) and (4) are repeatedly re-generated, so it is not correct to state that, for instance, $y_3(t) = y_1(t) + a_3t$.

The seasonality is simulated by adding another oscillating function whose oscillation frequency is less than ω :

$$y_5(t) = [a_1 + 0.25\Theta_9(T)]\sin(\omega t) + a_2\Theta_{10}(T) + [a_4 + 0.25\Theta_{11}(T)]\cos(\nu t), \quad (5)$$

$$y_6(t) = [a_1 + 0.25\Theta_{12}(T)]r(t) + a_2\Theta_{13}(T) + [a_4 + 0.25\Theta_{14}(T)]a_5 \cos(\nu t), \quad (6)$$

where $a_4 > 0$, $a_5 > 0$, and $\omega > \nu > 0$. Then the case of having simultaneously linear trend and seasonality is obtained by the addition [8], [15]:

$$y_7(t) = [a_1 + 0.25\Theta_{15}(T)]\sin(\omega t) + a_2\Theta_{16}(T) + a_3t + [a_4 + 0.25\Theta_{17}(T)]\cos(\nu t), \quad (7)$$

$$y_8(t) = [a_1 + 0.25\Theta_{18}(T)]r(t) + a_2\Theta_{19}(T) + a_3t + [a_4 + 0.25\Theta_{20}(T)]a_5 \cos(\nu t). \quad (8)$$

The process of exponential decay (implying a gradual decay of the time series magnitude) is simply defined by a rate $a_6 > 0$:

$$y_9(t) = [a_1 + 0.25\Theta_{21}(T)]\sin(\omega t)e^{-a_6t} + a_2\Theta_{22}(T), \quad (9)$$

$$y_{10}(t) = [a_1 + 0.25\Theta_{23}(T)]r(t)e^{-a_6t} + a_2\Theta_{24}(T). \quad (10)$$

The process when the time series magnitude is gradually rising is generated similarly:

$$y_{11}(t) = [a_1 + 0.25\Theta_{25}(T)]\sin(\omega t)e^{a_6t} + a_2\Theta_{26}(T), \quad (11)$$

$$y_{12}(t) = [a_1 + 0.25\Theta_{27}(T)]r(t)e^{a_6t} + a_2\Theta_{28}(T). \quad (12)$$

The vanishing (decaying) time series with a linear trend is obtained by just adding the trend to (9) and (10):

$$y_{13}(t) = [a_1 + 0.25\Theta_{29}(T)]\sin(\omega t)e^{-a_6t} + a_2\Theta_{30}(T) + a_3t, \quad (13)$$

$$y_{14}(t) = [a_1 + 0.25\Theta_{31}(T)]r(t)e^{-a_6t} + a_2\Theta_{32}(T) + a_3t. \quad (14)$$

The oscillating time series with exponential decay is obtained from (5) and (6), but the exponent is multiplied by both sine and cosine components:

$$y_{15}(t) = [a_1 + 0.25\Theta_{33}(T)]\sin(\omega t)e^{-a_6t} + a_2\Theta_{34}(T) + [a_4 + 0.25\Theta_{35}(T)]\cos(\nu t)e^{-a_6t}, \quad (15)$$

$$y_{16}(t) = [a_1 + 0.25\Theta_{36}(T)]r(t)e^{-a_6t} + a_2\Theta_{37}(T) + [a_4 + 0.25\Theta_{38}(T)]a_5 \cos(\nu t)e^{-a_6t}. \quad (16)$$

Obviously, the case of linear trend and seasonality with exponential decay is obtained by aggregating (13), (14) and (15), (16), respectively:

$$y_{17}(t) = [a_1 + 0.25\Theta_{39}(T)]\sin(\omega t)e^{-a_6t} + a_2\Theta_{40}(T)$$

$$+ a_3t + [a_4 + 0.25\Theta_{41}(T)]\cos(\nu t)e^{-a_6t}, \quad (17)$$

$$y_{18}(t) = [a_1 + 0.25\Theta_{42}(T)]r(t)e^{-a_6t} + a_2\Theta_{43}(T) + a_3t + [a_4 + 0.25\Theta_{44}(T)]a_5 \cos(\nu t)e^{-a_6t}. \quad (18)$$

The remaining cases with exponential growth are obtained directly from (13)–(18) by just inverting the sign of the exponential rate:

$$y_{19}(t) = [a_1 + 0.25\Theta_{45}(T)]\sin(\omega t)e^{a_6t} + a_2\Theta_{46}(T) + a_3t, \quad (19)$$

$$y_{20}(t) = [a_1 + 0.25\Theta_{47}(T)]r(t)e^{a_6t} + a_2\Theta_{48}(T) + a_3t, \quad (20)$$

$$y_{21}(t) = [a_1 + 0.25\Theta_{49}(T)]\sin(\omega t)e^{a_6t} + a_2\Theta_{50}(T) + [a_4 + 0.25\Theta_{51}(T)]\cos(\nu t)e^{a_6t}, \quad (21)$$

$$y_{22}(t) = [a_1 + 0.25\Theta_{52}(T)]r(t)e^{a_6t} + a_2\Theta_{53}(T) + [a_4 + 0.25\Theta_{54}(T)]a_5 \cos(\nu t)e^{a_6t}, \quad (22)$$

$$y_{23}(t) = [a_1 + 0.25\Theta_{55}(T)]\sin(\omega t)e^{a_6t} + a_2\Theta_{56}(T) + a_3t + [a_4 + 0.25\Theta_{57}(T)]\cos(\nu t)e^{a_6t}, \quad (23)$$

$$y_{24}(t) = [a_1 + 0.25\Theta_{58}(T)]r(t)e^{a_6t} + a_2\Theta_{59}(T) + a_3t + [a_4 + 0.25\Theta_{60}(T)]a_5 \cos(\nu t)e^{a_6t}. \quad (24)$$

An example of time series generated by (1)–(24) for $a_1 = 2$, $a_2 = 0.175$, $\omega = 0.14$, $a_3 = 0.01$, $a_4 = 5$, $a_5 = 0.18$, $\nu = 0.02$, $a_6 = 0.001$, $T = 1000$ is presented in Fig. 1. Every sequence $r(t)$ is of 10 subsequences (the right-side subplots): $P_r = 10$.

Relevant comparisons between benchmark instances are only possible using standardized (normalized) data. The available data are formally denoted by $\{y(t_i)\}_{i=1}^{T_0}$, where, without losing generality, $t_i = i$. So, T_0 is an amount of data to identify a model. If T_{forec} is a forecasting horizon (a time point beyond which forecasts are not made), where $T_0 < T_{\text{forec}} \leq T$, then data $\{y(t_i)\}_{i=T_0+1}^{T_{\text{forec}}}$ are used for testing the forecasting quality (performance). These data are normalized (standardized to the range from 0 to 1) as follows [3], [16]:

$$u(t_i) = \frac{y(t_i) - \min_{k=T_0+1, T_{\text{forec}}} y(t_k)}{\max_{k=T_0+1, T_{\text{forec}}} y(t_k) - \min_{k=T_0+1, T_{\text{forec}}} y(t_k)}. \quad (25)$$

If $\{\tilde{y}(t_i)\}_{i=T_0+1}^{T_{\text{forec}}}$ are forecast data, they are also normalized (with respect to the initial data):

$$\tilde{u}(t_i) = \frac{\tilde{y}(t_i) - \min_{k=T_0+1, T_{\text{forec}}} y(t_k)}{\max_{k=T_0+1, T_{\text{forec}}} y(t_k) - \min_{k=T_0+1, T_{\text{forec}}} y(t_k)}. \quad (26)$$

Fig. 1. A tiled set of 12 sine-shaped time series (left) and 12 random-like time series (right). The tiling fairly corresponds to the properties and tags in Table I.

Then the RMSE is

$$\rho_{\text{RMSE}} = \sqrt{\frac{1}{T_{\text{forec}} - T_0} \sum_{i=T_0+1}^{T_{\text{forec}}} [u(t_i) - \tilde{u}(t_i)]^2}. \quad (27)$$

The mean absolute error (MAE) [1], [5] calculated here as

$$\rho_{\text{MAE}} = \frac{1}{T_{\text{forec}} - T_0} \sum_{i=T_0+1}^{T_{\text{forec}}} |u(t_i) - \tilde{u}(t_i)|$$

is another measure of forecasting accuracy. However, the difference between RMSE and MAE is not that much. MAE is sometimes claimed to be more convenient (suitable) but reasons behind this are quite unclear. Obviously, owing to the square, RMSE intensifies greater errors. A drawback of both RMSE and MAE is that they do not show outliers. This is so because outliers, if any, are lost due to averaging. Therefore, the maximum absolute error (MaxAE) should be calculated instead of MAE [17]:

$$\rho_{\text{MaxAE}} = \max_{i=T_0+1, T_{\text{forec}}} |u(t_i) - \tilde{u}(t_i)|. \quad (28)$$

MaxAE (28) registers information about the worst outlier. Eventually, RMSE (27) and MaxAE (28) will be used to see the averaged and worst errors in forecasting by different methods. Nevertheless, to see the performance while the forecasting horizon is changed, non-normalized RMSE

$$\lambda_{\text{RMSE}} = \sqrt{\frac{1}{T_{\text{forec}} - T_0} \sum_{i=T_0+1}^{T_{\text{forec}}} [y(t_i) - \tilde{y}(t_i)]^2} \quad (29)$$

and MaxAE

$$\lambda_{\text{MaxAE}} = \max_{i=T_0+1, T_{\text{forec}}} |y(t_i) - \tilde{y}(t_i)| \quad (30)$$

can be used as well. RMSE (29) and MaxAE (30) allow selecting the most appropriate horizon, at which a given method performs best.

IV. BENCHMARK ANALYSIS

Let us consider the following example. Given 840 observations of a sine-shaped series (the one in the upper left subplot in Fig. 1), the task is to forecast next 840 observations. An ARIMA-based model is obtained by setting the NSARP degree at 10 and NSI degree at 1. The result is visibly unacceptable (Fig. 2). When polyline $y_1(t)$ is smoothed by a moving average filter with the 2-neighbor span, and the ARIMA model is applied to the smoothed time series, the result does not change much (Fig. 3), although the first sine half-period is forecasted rather well. Forecasts by an LSTMN are far from perfect (Fig. 4). Therefore, a simple question arises: might the amount in 840 observations be insufficient to make acceptable forecasts on such a trivial time series shape?

Fig. 2. The sine-shaped series (see Fig. 1) and its quite unacceptable forecast.

Fig. 3. The smoothed sine-shaped series (see Fig. 1) and its still unacceptable forecast. However, the forecast is not that poor if the horizon is less than 880.

Fig. 4. The forecasts for the initial (top) and smoothed (bottom) sine-shaped series (see Fig. 1) by an LSTMN with a single LSTM layer in 32 neurons. LSTMNs with either more layers or neurons do not perform better.

The obvious answer is no, it is sufficient, but it may be superfluous. If $y_1(t)$ is downsampled with a factor s , the ARIMA model performs much better (Fig. 5) except for $s = 2$, where final forecasts are interpolated from the forecasts on the downsampled time series [18], [19]. The LSTMN performance is also improved by downsampling (Fig. 6).

Fig. 5. The ARIMA forecasts by downsampling the sine-shaped series.

Fig. 6. The LSTMN forecasts by downsampling the sine-shaped series.

It is noteworthy that smoothing and downsampling in this simplest example has a very positive impact on ARIMA forecasts. Owing to smoothing the sine-shaped time series, which is downsampled afterwards, the performance is significantly improved: RMSE (27) does not exceed 0.048 and MaxAE (28) does not exceed 0.2 for $s \in \{4, 6, 8, 10\}$. It is a significant numerical and visual (compare Fig. 2 and Fig. 3 to Fig. 5) improvement of forecasting. Without smoothing, the downsampled time series is forecast worse: RMSE (27) varies between 0.054 ($s = 6$) and 0.094 ($s = 8$) for $s \in \{4, 6, 8, 10\}$, whereas MaxAE (28) varies between 0.18 ($s = 4$) and 0.248 ($s = 8$).

Despite smoothing in Fig. 4 “loses” to straightforward (raw) forecasting, the smoothing-and-downsampling approach has a positive impact on LSTMN forecasts also. At it is seen from Fig. 6, downsampling the smoothed series with factor $s = 10$ allows obtaining rather good forecasts by RMSE (27) and MaxAE (28) not exceeding 0.069 and 0.2, respectively (the forecasts by $s = 12$ are less accurate due to the MaxAE exceeds 0.25). Meanwhile, the best accuracy forecasts by downsampling the raw series are obtained by $s = 2$ with the RMSE (27) and MaxAE (28) not exceeding 0.047 and 0.223, respectively. This result only seems to be better than that for the smoothed series. Truly, the best (i.e., minimal) MaxAE of the downsampled raw series is 11.5 % greater than the best MaxAE of the downsampled smoothed series. Overall, LSTMN forecasts are less accurate than ARIMA forecasts (compare Fig. 5 to Fig. 6) for the sine-shaped series, whether it is smoothed or not.

After all, the sine-shaped series may be treated as the most trivial benchmark. Its triviality may distort influence of downsampling in times series more common for real practice. Therefore, let us consider another, more complicated time series. This is the sine-shaped series with seasonality, which is modelled by (5). The ARIMA forecasts for this time series (see the third subplot from the top in the left column in Fig. 1) are presented in Fig. 7. It is quite clear (see the first two subplots from the top) that the task of forecasting by ARIMA is unrealisable without downsampling. The results obtained by $s = 2$ and $s = 4$ are also unacceptable. Downsampling with $s = 6$ gives a better forecasting accuracy but still the 840 observations are not forecast, although the first 10 to 25 forecasts are relatively accurate. At last, for $s = 8$ the shape of the forecast curve (interpolated from the forecasts on the downsampled time series) really resembles the sine-shaped series with seasonality. Downsampling with $s = 12$ (where only every 12th observation is processed) gives the best forecasts by $\rho_{RMSE} = 0.027598$ and $\rho_{MaxAE} = 0.14115$. Further downsampling starts increasing both RMSE and MaxAE.

Fig. 7. The ARIMA forecasts of the sine-shaped series with seasonality.

The best ARIMA result by raw forecasting for the sine-shaped series with seasonality is also obtained by $s=12$, where $\rho_{\text{RMSE}} = 0.060169$ and $\rho_{\text{MaxAE}} = 0.17304$. This accuracy is hardly comparable to that in the case of smoothing. Is it so for LSTMN? Indeed, it is. The best LSTMN result by the smoothing-and-downsampling approach is achieved by $s=12$ (Fig. 8), where the forecasting accuracy (with $\rho_{\text{RMSE}} = 0.036362$ and $\rho_{\text{MaxAE}} = 0.11126$) is almost the same as the ARIMA accuracy. If the forecasts are made without smoothing, the best accuracy is achieved by $s=4$, where $\rho_{\text{RMSE}} = 0.074597$ and $\rho_{\text{MaxAE}} = 0.22611$.

Fig. 8. The LSTMN forecasts of the sine-shaped series with seasonality.

Special attention must be paid to those forecast “tails” seen in bottom subplots in Figs. 5–8, where forecasts are like bursts nearer to the forecasting horizon. The “bursting tails” are formed as a result of interpolation after downsampling with $s=14$ (where only every 14th observation is processed). A smaller “bursting tail” is formed by downsampling the sine-shaped series with $s=12$ (Fig. 5). Apparently, RMSE (27) and

MaxAE (28) by $s=14$ increase due to such “bursting tails”. Could the RMSE and MaxAE be made less by cutting those “tails” off? This question is answered by varying the forecasting horizon from the shortest (let it be 10 time points) up to the longest (the whole test set of 840 points in this case) with a step of 10 points. RMSE (29) for the raw (Fig. 9) and smoothed (Fig. 10) sine-shaped series with seasonality confirm that cutting off the “tails” makes the forecasting more accurate even by deeper downsampling. The difference is especially significant for the case with smoothing the series (see Fig. 10), wherein deeper downsampling strongly outperforms downsampling with $s=12$. However, the difference between MaxAE (30) for the raw (Fig. 11) and smoothed (Fig. 12) sine-shaped series with seasonality is not that simple. It is hard to determine only with the MaxAE that $s=14$ improves the forecasting accuracy.

Fig. 9. RMSE (29) of the ARIMA forecasts for the raw sine-shaped series with seasonality (Fig. 7). The last 10 points (the “bursting tail”) are cut off. Deeper downsampling (with $s=14$) is not worse than downsampling with $s=12$ by forecasting no more than 320 points. When the horizon is longer, deeper downsampling becomes better (i.e., it outperforms downsampling with $s=12$). In other words, the forecasting accuracy is improved by processing even fewer number of observations.

Fig. 10. RMSE (29) of the ARIMA forecasts for the smoothed sine-shaped series with seasonality (Fig. 7) without the “bursting tail” cutoff. Both the RMSE are less than those in Fig. 9 (in the case without smoothing the series). After smoothing, deeper downsampling (with $s=14$) is better than downsampling with $s=12$ for any forecasting horizon.

Fig. 11. MaxAE (30) of the ARIMA forecasts for the raw sine-shaped series with seasonality without the “bursting tail” offset. Deeper downsampling (with $s = 14$) outperforms downsampling with $s = 12$ by the forecasting horizon length of 100 points and longer.

Fig. 12. MaxAE (30) of the ARIMA forecasts for the smoothed sine-shaped series with seasonality without the “bursting tail” offset. After smoothing, deeper downsampling (with $s = 14$) outperforms downsampling with $s = 12$ for very short (by forecasting no more than 40 points) and long (by forecasting no less than 750 points) forecasting horizons.

RMSE (27) and (29) of the best ARIMA forecasts for each type of time series (Table I) by $P_r = 7$ are presented in Fig. 13.

Fig. 13. RMSE (27) and (29) of the best ARIMA forecasts (see Tables I and II).

Owing to RMSE (29), it is clear that smoothing time series improves its forecasting accuracy. The respective options of smoothing and downsampling are disclosed in Table II. Meanwhile, MaxAE (28) and (30) do not show the improving property of the smoothing (see Fig. 14 and Table III). It is so probably due to the number of IRSS is decreased. RMSE (27) and (29) of the LSTMN forecasts (see Fig. 15 and Table IV) do not show that either. It rather seems that the smoothing does not improve LSTMN forecasts. Eventually, this is confirmed by MaxAE (28) and (30) in Fig. 16 (see Table V).

Fig. 14. MaxAE (28) and (30) of the ARIMA forecasts (see Tables I and III).

Fig. 15. RMSE (27) and (29) of the LSTMN forecasts (see Tables I and IV).

Fig. 16. MaxAE (28) and (30) of the LSTMN forecasts (see Tables I and V).

TABLE II
SMOOTHING (BOLD) AND DOWNSAMPLING (S) OPTIONS OF THE BEST RMSE IN THE ARIMA FORECASTS (FIG. 13)

Tag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
ρ_{RMSE}	6	14	10	2	12	none	14	none	4	12	10	10	10	none	14	12	4	none	10	2	12	none	12	none
ρ_{RMSE}	6	14	12	none	12	10	14	4	10	12	10	10	6	none	14	8	14	8	10	12	10	10	10	none
λ_{RMSE}	6	14	4	6	4	12	14	2	none	none	6	10	14	2	14	12	4	6	2	14	6	8	6	6
λ_{RMSE}	12	8	4	6	10	4	4	none	4	8	4	12	6	8	4	none	4	4	10	12	10	none	4	none

TABLE III
SMOOTHING (BOLD) AND DOWNSAMPLING (S) OPTIONS OF THE BEST MAXAE IN THE ARIMA FORECASTS (FIG. 14)

Tag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
ρ_{MaxAE}	4	14	10	none	12	none	14	none	4	10	10	10	10	none	12	12	4	8	6	none	12	10	12	none
ρ_{MaxAE}	10	10	6	none	12	10	12	4	10	12	10	10	8	none	14	8	14	8	6	2	10	10	10	none
λ_{MaxAE}	12	none	4	6	4	12	14	none	none	none	12	6	2	12	8	12	12	4	12	2	14	6	2	6
λ_{MaxAE}	12	14	4	6	14	4	4	none	4	2	4	8	12	8	4	4	4	2	4	12	10	6	4	none

TABLE IV
SMOOTHING (BOLD) AND DOWNSAMPLING (S) OPTIONS OF THE BEST RMSE IN THE LSTMN FORECASTS (FIG. 15)

Tag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
ρ_{RMSE}	6	8	none	2	8	14	none	4	10	8	10	10	14	2	12	12	2	6	10	12	8	none	2	none
ρ_{RMSE}	12	8	none	2	14	12	2	6	10	12	10	10	12	none	14	10	14	2	14	10	12	12	8	none
λ_{RMSE}	4	4	8	none	8	10	12	2	2	14	8	6	12	2	6	none	2	6	none	6	12	none	2	none
λ_{RMSE}	12	12	6	6	14	10	4	none	14	14	2	10	2	6	4	14	14	2	4	none	14	10	4	4

TABLE V
SMOOTHING (BOLD) AND DOWNSAMPLING (S) OPTIONS OF THE BEST MAXAE IN THE LSTMN FORECASTS (FIG. 16)

Tag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
ρ_{MaxAE}	6	8	none	10	6	8	none	4	10	4	10	10	10	4	12	12	2	2	10	12	8	14	2	none
ρ_{MaxAE}	10	4	none	2	14	12	2	6	10	12	12	10	12	none	14	14	12	2	10	10	12	14	10	2
λ_{MaxAE}	4	10	4	2	8	10	2	2	2	none	8	6	12	2	6	2	2	6	none	6	12	none	2	none
λ_{MaxAE}	12	2	6	6	4	10	4	none	2	2	2	10	2	6	4	14	14	2	2	none	8	10	4	4

The non-bold word “none” in Tables II–V (the 1st and 3rd rows) means that the best forecasting is obtained by neither smoothing nor downsampling. The bold word “none” (the 2nd and 4th rows standing for smoothing) means that the best forecasting is obtained by smoothing without downsampling. For instance, the word “none” in the 2nd row of Table II for a linear-trend time series (the 4th column) means that the respective RMSE by (27) has been obtained minimal by smoothing without downsampling. The same result has been obtained for MaxAE (28) in Table III. As it is clearly seen from Tables II–V, the number of occurrences when the best forecasting is obtained by neither smoothing nor downsampling is less than the total number of options to smooth and downsample. In fact, it is 34 (the total number of non-bold words “none” in Tables II–V) out of 384 instances (see Table VI). Therefore, it can be asserted that only about 9 % of the best performance has been achieved without the smoothing-and-downsampling approach (when neither smoothing nor downsampling can improve the forecasting accuracy).

Apparently, the smoothing does not always increase the accuracy. Table VI shows that, if the time series is not downsampled, it is just about 38 % possibility of that the smoothing will allow obtaining better accuracy. Even when the time series is downsampled, the preliminary smoothing is not

always recommended. Thus, when the raw time series is downsampled with $s \in \{2, 6, 8, 12\}$, altogether 98 occurrences of the best forecasting accuracy have been obtained versus 77 occurrences of the smoothed time series. When the raw time series is downsampled with $s \in \{4, 10, 14\}$, it is 60 occurrences of the best forecasting accuracy versus 94 occurrences of the smoothed time series. So, before downsampling, 158 best occurrences of raw time series have been obtained versus 171 best occurrences of the smoothed time series.

Table VI
OCCURRENCES OF THE BEST FORECASTING ACCURACY

Downsampling factor	Number of occurrences when either RMSE or MaxAE by either ARIMA or LSTMN forecasts are obtained the best		
	Raw series (192 instances)	Smoothed series (192 instances)	Total (384 instances)
none	34	21	55
2	28	20	48
4	19	32	51
6	24	16	40
8	15	13	28
10	25	38	63
12	31	28	59
14	16	24	40

Nevertheless, downsampling itself is far more useful than smoothing. In just 55 best occurrences (about 14.3 %), downsampling the time series has made forecasting accuracy worse. An example of a successful application of the smoothing-and-downsampling approach is presented in Fig. 17,

where a visibly hard-to-get-it time series is smoothed and downsampled with $s=10$, which eventually allows obtaining an acceptable forecasting accuracy (apart from the RMSE or MaxAE in the bottom subplot, the forecasting is visibly of high quality).

Fig. 17. The best ARIMA forecast of a sequence of 7 IRSS having seasonality with exponential decay modelled by (16). Although only every 10th point is processed by ARIMA, the resulting forecasts are interpolated and the interpolation fits closely the initial raw sequence. Surely, a perfect match is impossible.

As the noise strengthens, the smoothing-and-downsampling approach becomes more applicable ensuring higher accuracy. Thus, testing the benchmark series by plugging $a_2 = 0.35$ into (1)–(24) confirms further robustness of downsampling: in just 50 best occurrences (about 13 %) downsampling the time series has made forecasting accuracy worse. Testing by varying other parameters of (1)–(24) roughly repeats the averaged better performance of the approach.

V. DISCUSSION

The obtained accuracy in the visualised examples may be far from optimal because the ARIMA and LSTMN are not optimised in this study. Moreover, testing various LSTMNs has not shown that LSTMNs forecast better than ARIMA. Eventually, truly optimal forecasting will probably take a lot of resources (e. g., for optimising ARIMA or LSTMN parameters).

Time series researchers should be aware of promising “reports” from bloggers, which sometimes do not distinguish the forecast horizon from forecasting repeatability. Changing the test set must not change the forecasts. Otherwise, the forecasting model has a fallacy.

Obviously, long-horizon forecasts cannot accurately follow every peak and pit in a test set. This is only possible when just one or few time steps are forecasted (a very short horizon), and the forecasting model is applied multiple times. In real-world practice, long-horizon forecasts are of much greater interest than short-horizon forecasts because they allow appropriately

making decisions and adjustments in the system whose time series is studied.

VI. CONCLUSION

Based on the performed analysis of the benchmark of 24 time series models, it is ascertained that, for improving the quality of forecasting, the time series should be smoothed and then downsampled. These operations can be fulfilled successively until the improvement fails. If preliminary smoothing worsens forecasts, the raw time series is straightforwardly downsampled until the forecasting accuracy by either RMSE or MaxAE starts dropping. However, if the time series has a visible property of being noised, the preliminary smoothing is strongly recommended.

REFERENCES

- [1] B. Schelter, M. Winterhalter, and J. Timmer, Eds., *Handbook of Time Series Analysis: Recent Theoretical Developments and Applications*. Wiley, 2006. <https://doi.org/10.1002/9783527609970>
- [2] V. Kotu and B. Deshpande, “Chapter 10 – Time Series Forecasting,” in *Predictive Analytics and Data Mining*, V. Kotu, B. Deshpande, Eds. Morgan Kaufmann, 2015, pp. 305–327. <https://doi.org/10.1016/B978-0-12-801460-8.00010-0>
- [3] J. G. De Gooijer and R. J. Hyndman, “25 Years of Time Series Forecasting,” *International Journal of Forecasting*, vol. 22, iss. 3, pp. 443–473, 2006. <https://doi.org/10.1016/j.ijforecast.2006.01.001>
- [4] C. Villaseñor, “Chapter 2 – Hyperellipsoidal Neural Network Trained With Extended Kalman Filter for Forecasting of Time Series,” in *Artificial Neural Networks for Engineering Applications*, A. Y. Alanis, N. Arana-Daniel, C. Lyppez-Franco, Eds. Academic Press, 2019, pp. 9–19. <https://doi.org/10.1016/B978-0-12-818247-5.00011-3>

- [5] V. Kotu and B. Deshpande, "Chapter 12 – Time Series Forecasting," in *Data Science (Second Edition)*, V. Kotu, B. Deshpande, Eds. Morgan Kaufmann, 2019, pp. 395–445.
<https://doi.org/10.1016/B978-0-12-814761-0.00012-5>
- [6] M. Fakhfekh and A. Jeribi, "Volatility Dynamics of Crypto-Currencies' Returns: Evidence from Asymmetric and Long Memory GARCH Models," *Research in International Business and Finance*, vol. 51, 101075, 2020. <https://doi.org/10.1016/j.ribaf.2019.101075>
- [7] F. C. Palm, "Chapter 7 – GARCH Models of Volatility," in *Handbook of Statistics*, vol. 14. Elsevier, 1996, pp. 209–240.
[https://doi.org/10.1016/S0169-7161\(96\)14009-8](https://doi.org/10.1016/S0169-7161(96)14009-8)
- [8] E. Ghysels, D. R. Osborn, and P. M. M. Rodrigues, "Chapter 13 – Forecasting Seasonal Time Series," in *Handbook of Economic Forecasting*, vol. 1, G. Elliott, C. W. J. Granger, A. Timmermann, Eds. Elsevier, 2006, pp. 659–711.
[https://doi.org/10.1016/S1574-0706\(05\)01013-X](https://doi.org/10.1016/S1574-0706(05)01013-X)
- [9] H. Shimodaira, "Chapter 36 – Time-Series Prediction," in *Expert Systems*, C. T. Leondes, Ed. Academic Press, 2002, pp. 1259–1313.
<https://doi.org/10.1016/B978-012443880-4/50080-6>
- [10] R. DiPietro and G. D. Hager, "Chapter 21 – Deep learning: RNNs and LSTM," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, S. K. Zhou, D. Rueckert, G. Fichtinger, Eds. Academic Press, 2020, pp. 503–519.
<https://doi.org/10.1016/B978-0-12-816176-0.00026-0>
- [11] T. Masters, "Chapter 4 – Time-Series Prediction," in *Practical Neural Network Recipes in C++*, T. Masters, Ed. Morgan Kaufmann, 1993, pp. 47–66. <https://doi.org/10.1016/B978-0-08-051433-8.50009-4>
- [12] M. Sangiorgio and F. Dercole, "Robustness of LSTM Neural Networks for Multi-Step Forecasting of Chaotic Time Series," *Chaos, Solitons & Fractals*, vol. 139, 110045, 2020.
<https://doi.org/10.1016/j.chaos.2020.110045>
- [13] R. Kneusel, *Random Numbers and Computers*. Springer International Publishing, 2018. <https://doi.org/10.1007/978-3-319-77697-2>
- [14] B. Quenneville and C. Gagné, "Testing Time Series Data Compatibility for Benchmarking," *International Journal of Forecasting*, vol. 29, iss. 4, pp. 754–766, 2013. <https://doi.org/10.1016/j.ijforecast.2011.10.001>
- [15] A. Stepchenko, J. Chizhov, L. Aleksejeva, and J. Tolujew, "Nonlinear, Non-Stationary and Seasonal Time Series Forecasting Using Different Methods Coupled with Data Preprocessing," *Procedia Computer Science*, vol. 104, pp. 578–585, 2017.
<https://doi.org/10.1016/j.procs.2017.01.175>
- [16] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Springer, Cham, 2016.
<https://doi.org/10.1007/978-3-319-29854-2>
- [17] R. E. Edwards, *Functional Analysis. Theory and Applications*. Hold, Rinehart and Winston, 1965.
- [18] A. Jeffrey, "Chapter 27 – Numerical Approximation," in *Handbook of Mathematical Formulas and Integrals*, 3rd ed., A. Jeffrey, Ed. Academic Press, 2004, pp. 409–417.
<https://doi.org/10.1016/B978-012382256-7/50030-0>
- [19] W. R. Madych, "Error Estimates for Interpolation by Generalized Splines," in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, L. L. Schumaker, Eds. Academic Press, 1991, pp. 297–306.
<https://doi.org/10.1016/B978-0-12-438660-0.50047-9>

Vadim V. Romanuke was born in 1979. He graduated from the Technological University of Podillya in 2001. In 2006, he received the Degree of Candidate of Technical Sciences in Mathematical Modelling and Computational Methods. The Candidate Dissertation suggested a way of increasing interference noise immunity of data transferred over radio systems. The degree of Doctor of Technical Sciences in Mathematical Modelling and Computational Methods was received in 2014. The Doctor-of-Science Dissertation solved a problem of increasing efficiency of identification of models for multistage technical control and run-in under multivariate uncertainties of their parameters and relationships. In 2016, he received the status of a Full Professor. He is a Professor of the Faculty of Mechanical and Electrical Engineering at the Polish Naval Academy. His current research interests concern decision making, game theory, semantic image segmentation, statistical approximation, and control engineering based on statistical correspondence. He has 375 published scientific articles, one monograph, one tutorial, four methodical guidelines in functional analysis, mathematical and computer modelling Master Thesis preparation, conflict-controlled systems, Master academic practice. Before January 2018, Vadim Romanuke was the scientific supervisor of a Ukrainian budget grant work concerning minimisation of water heat transfer and consumption.

Address for correspondence: 69 Śmidowicza Street, Gdynia, Poland, 81-127.

E-mail: romanukevadimv@gmail.com

ORCID iD: <https://orcid.org/0000-0003-3543-3087>