NIST Technical Note 2218

DeepFit: automated distribution fitting for building stochastic models

Siham Khoussi
Alan Heckert
Abdella Battou
Saddek Bensalem

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2218

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Technical Note 2218

DeepFit: automated distribution fitting for building stochastic models

Siham Khoussi
Communications Technology Laboratory

Alan Heckert
Statistical Engineering Division
Information Technology Laboratory

Abdella Battou
Communications Technology Laboratory

Saddek Bensalem
University of Grenoble Alpes (UGA)
Grenoble, France

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2218

April 2022

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Undersecretary of Commerce for Standards and Technology

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Technical Note 2218
Natl. Inst. Stand. Technol. Tech. Note 2218, 20 pages (April 2022)
CODEN: NTNOEF

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2218

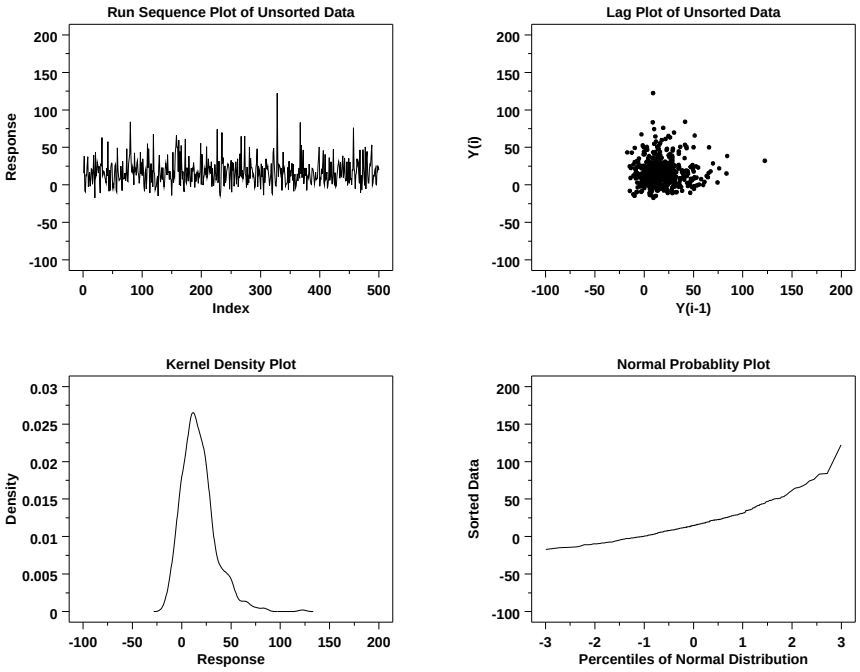

Fig. 4. 4-plot of 500 random gumbel-max points.

2.2 Neural networks classification

In this module, an initial transformation is applied to the input dataset then it's fed to the neural network classifier which was trained on a large database consisting of commonly used distributions and continuous measurements where the data are not binned, censored or truncated. This classifier takes as input a kernel density plot of the data to fit and makes a prediction on the best fit distribution model for it. We refer the reader to our study [9] for further details on the trained model. We experimented with several transformation algorithms but, only two yielded promising results, that is the kernel density normalization and the u-score normalization:

1. The u-score normalization, also referred to as the Min-Max scaler, transforms the observations to a (0,1) scale according to the following mathematical formulation:

$$u\_score = \frac{x - min(x)}{max(x)) - min(x)} \tag{1}$$

   x is the original observation value, u_score is the normalized value, $min(x)$ and $max(x)$ are respectively the minimum and maximum values observed in this data.

2. The kernel density normalization transforms the kernel density heights to integrate to 1 on the 1 to 256 x-coordinate scale

$$k\_score = \frac{x}{\sum_{i=1}^{256} x_i} \tag{2}$$

where x is the original value and k_score is the normalized value.

In our study [9], we found that these two techniques are non-distorting of the shape of the kernel density plot and preserve the form of the probability distribution regardless of the location and scale values

2.3 Parameter estimation

Once the neural network classifier has identified the "best" distributional model, the parameters of the distribution are estimated in this module via the maximum likelihood method [5]. This module estimates three different output sections:

1. Some basic summary statistics for the observations (e.g. minimum, maximum, range, skewness, kurtosis, etc.).

2. The parameter estimates (location and scale). Note that, we are continuously adding more distributions to the tool in order to support the ones with the shape parameters.

3. Confidence intervals for the estimated parameters.

2.4 Evaluation

This module includes traditional statistical goodness of fit techniques to determine if the distribution model suggested by the neural networks is in fact appropriate for the data. Generally, there are three basic categories of the goodness of fit tests:

1. The first category is based on comparing the empirical cumulative distribution function (CDF) (i.e., based on the data) to the theoretical CDF function. This includes tests such as the Kolmogorov-Smirnov (KS) test, the Anderson-Darling (AD).

2. The second category relies on the percent point function (PPF). Tests in this category compare the differences between the empirical PPF to the theoretical PPF. This includes the probability plot correlation coefficient test (PPCC) [11].

3. The third category relies on the likelihood function. As the name "maximum likelihood" implies, this module searches for the distribution that provides the maximum value of the likelihood function. Note that, it is more common to use "information critieria" which is also based on the value of the likelihood function. Examples of this category include the Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC). The latter is more commonly used.

Currently DeepFit includes four goodness of fit statistics from the categories described above: Kolmogorov-Smirnov (KS), Anderson Darling (AD), PPCC and the BIC information criterion. This choice is justified by the fact that AD and KS tests are more powerful for the type of distributions supported by the neural network classifier at the moment[2]. We also suggest the analyst to follow up with the probability plot correlation coefficient test (PPCC) from the second category. For the list of commonly used distributions considered in this study, using BIC is equivalent to just using the likelihood value since all the distributions have the same number of parameters. However, it will be more useful as we continue supporting additional distributions with one or more shape parameters in the neural networks models.

2.5 Best Fit ranking

This module uses several goodness of fit tests to rank the supported distributions from the best match to the last match.

3. Tool assessment

DeepFit was evaluated on synthetic data [9] and real-world data [12] and successfully modeled several commonly used distributions. In this section, we use one example of real measurements obtained from a published study on Heat Flow Meter Calibration & Stability Analysis [12] to demonstrate the functionalities of DeepFit.

Figure 5 presents the 4-plots method implemented in the first module of DeepFit (i.e., data screening). The first two plots (i.e., the run sequence and the lag plots) show no obvious trends in the data which indicates that this dataset comes from a random process. Additionally, the kernel density plot looks symmetric and the normal probability plot is linear. This suggests that the normal probability distribution is probably a good fit for this data. In Figure 6, the pre-screened data is normalized by selecting one of the two methods: the u-score and the kernel normalization methods before it is passed through the neural networks classifier. The latter predicts the best candidate model for the data from the currently supported distributions. In this example, the normal distribution was selected which corresponds to the initial assumption made in the first module (Figure 5). Next is the parameter estimation module as shown in Figure 7. In this module, the parameters of the distribution that was previously selected by the neural networks classifier are estimated. That is, the location, the scale and the shape[3] parameters as well as their associated confidence intervals. For this example, the location and scale parameters are estimated for the normal probability distribution as in Figure 7. In this step, the analyst can also generate random samples from the selected distribution and store it locally into a proper format or plot both the original

[2]Uniform, normal, logistic, exponential, half-normal, half-logistic, double-exponential, gumbel-max and gumbel-min

[3]Currently not supporting families of distribution with one or more shape parameters

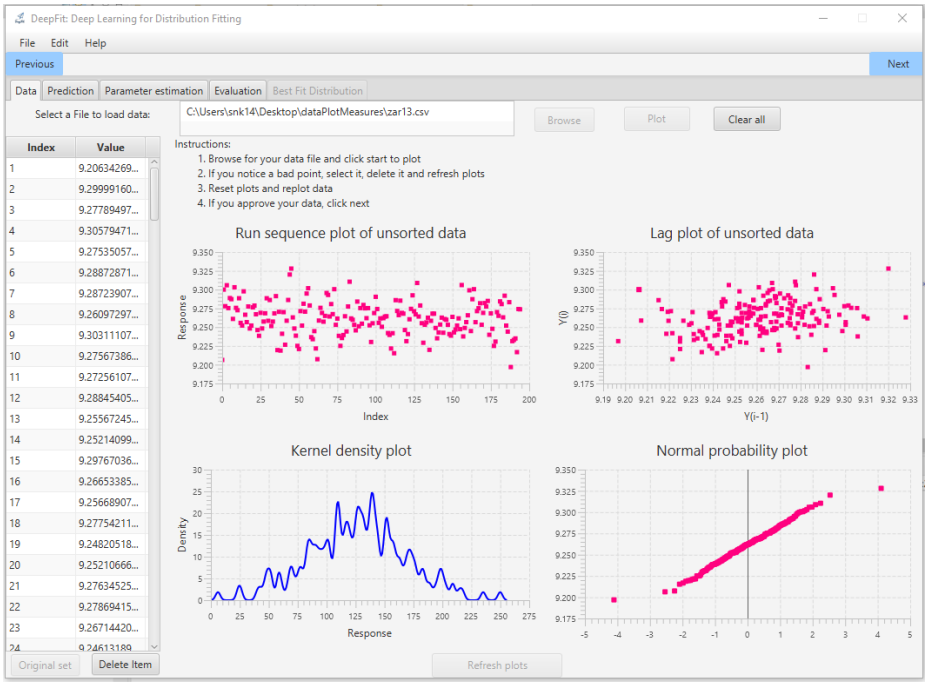

Fig. 5. Module 1: Data screening

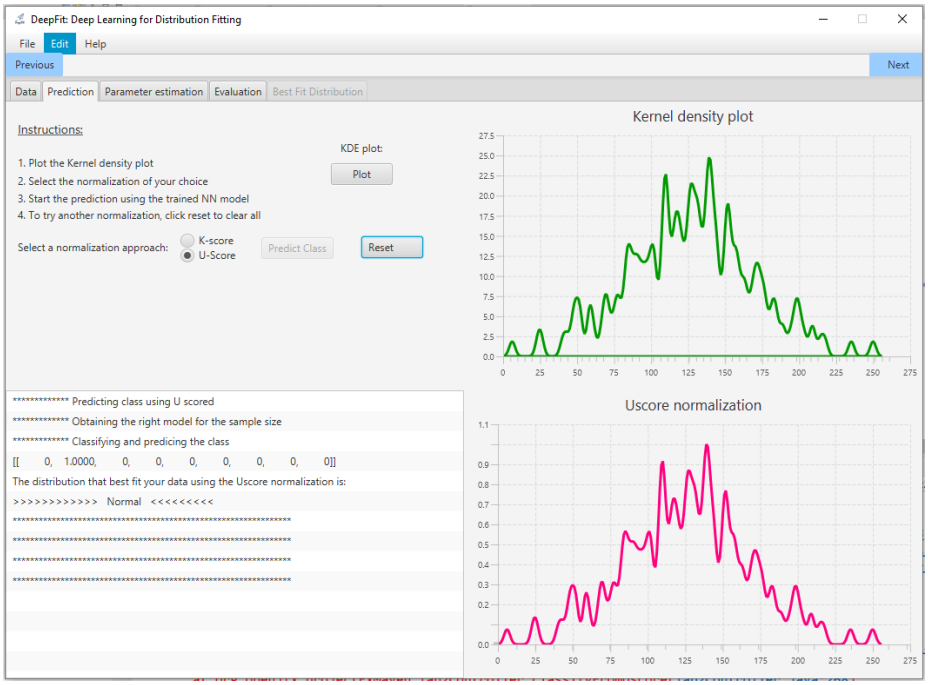

Fig. 6. Module 2: Neural networks classification

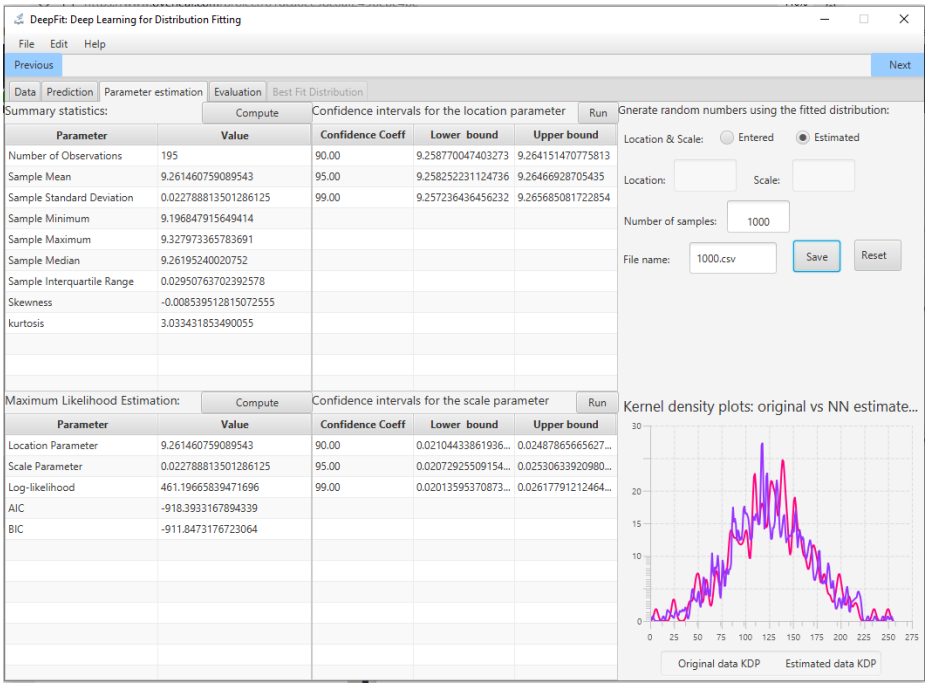

Fig. 7. Module 3: Parameter estimation

kdp of the input data and the kdp of N random samples generated from the NN estimation to see if they are similar. Figure 7 indicates that both plots look almost identical.

For further assessment, the analyst might choose to run several goodness of fit tests, supported in DeepFit, to confirm the neural networks classification or to test different distributions from the provided drop-down menu as in Figure 8. The currently supported goodness of fit tests include the Anderson Darling (AD), the Kolmogorov–Smirnov (KS) and the probability plot confidence coefficient (PPCC) tests. In certain cases, these tests could reach different conclusions because each of them is evaluating specific features of the data. As an example, the AD, KS and PPCC can be sensitive to different types of departures from the hypothesized distributions (e.g., AD is more sensitive to differences in the tails unlike the KS test, the PPCC test is a lower tailed test).

Finally, Figure 9 presents the fifth module of the tool which includes the option to bypass the neural networks and simply run a few goodness of fit tests to rank the supported distributions from the best match to the last. When applied to the Heat Flow Meter Calibration & Stability Analysis dataset, the normal distribution ranked first which matches the NN prediction made in module 2 (Figure 6).

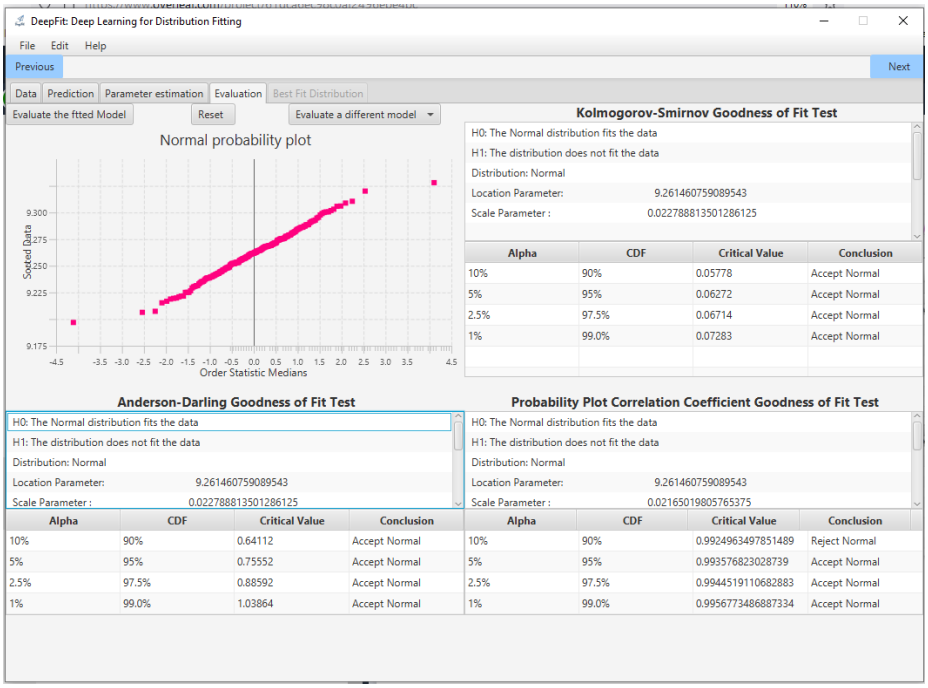

Fig. 8. Module 4: Evaluation

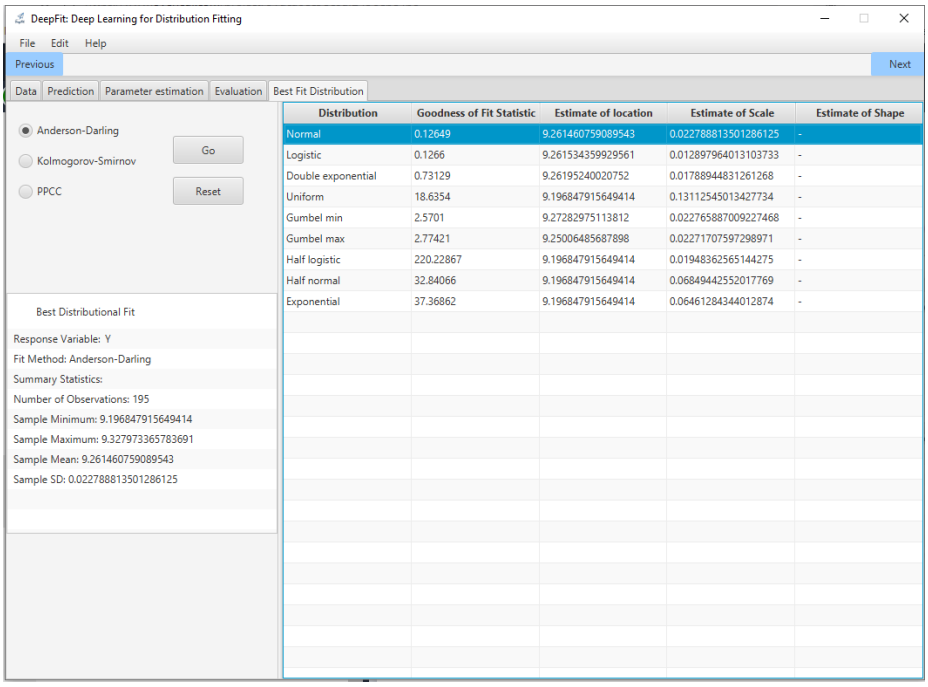

Fig. 9. Module 5: Best fit ranking

4. SMC context

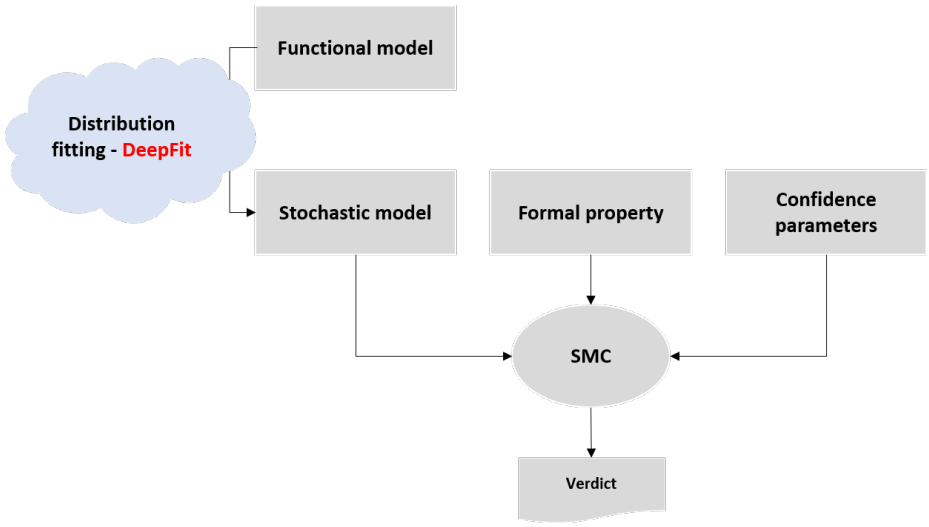

Fig. 10. DeepFit in the context of SMC

In this paper, we propose a tool for data analysis to be used in the context of SMC verification. SMC takes as input a stochastic model a property to verify and a set of confidence parameters to control the accuracy of the evaluation. The stochastic model is obtained by calibrating the functional behavior of a system with probabilistic variables, which are updated via probability distributions (PD). A PD is, typically, obtained by collecting and analyzing measurements from the system's execution using traditional statistical tests to select the best fit distribution (i.e., distribution fitting). Distribution fitting is crucial for the correct assessment via SMC and it's an important preliminary step in science and engineering, in general. However, this task requires a good statistical background and familiarity with several distributions which is beyond the expertise of some analysts. Therefore, we propose to use DeepFit, to automate the distribution fitting process as part of the workflow presented in Figure 10.

5. Lessons learned

In this paper, we propose DeepFit, a combined effort between neural networks and statistical techniques for data analysis and distributional fitting. DeepFit provides a preliminary step of data screening to remove bad data points (e.g., data is mis-coded or there is an assignable cause for why the observation is in error). Then uses a neural networks classifier that was previously trained on a large set of commonly used distributions in order to select the 'best' candidate model given a set of empirical observations. Moreover, the tool incorporates a variety of traditional statistics designed to compute the parameters of the selected distribution as well as assess it's goodness of fit. Additionally, DeepFit has

the advantage of being used as a standalone tool for fitting data or can be included in the workflow of SMC which was the initial motivation behind this work. We explained that one of the inputs to SMC is the stochastic model for the system to verify which is obtained by calibrating the functional model with probabilistic variables that are updated via probability distributions (PD). A PD is obtained by analyzing measurements from the system's execution using traditional statistical tests via a process called distribution fitting. These tests generally require a deeper understanding and familiarity with many probability distributions in order to interpret their numerical and graphical outputs. However, some analysts aren't equipped with such statistical background and are at risk of making faulty judgments or uneducated guesses of the underlying distribution from the data, hence leading to incorrect verification of the system via SMC. As such, we suggest to use our tool in this context to automate the distribution fitting process.[4]

6. Future work

Currently, DeepFit[5] supports a limited list of commonly used distributions in science and engineering to serve as a proof of concept of the viability of our approach. In the future, we plan to extend the number of supported distributions to also include families of distributions, such as the Weibull, Lognormal, Gamma distributions and other use cases as well as incorporate the ability to make more specific classifications (e.g., distinguish between Weibull or Lognormal) and compare this to approaches such as the likelihood ratio test [13], [14]. We also plan to explore a different type of neural networks such as the Long Short-Term Memory (LSTM) networks [15] [16]. These networks are a type of recurrent neural network, with the ability to learn order dependence in sequence prediction problems.

References

[1] Legay A, Lukina A, Traonouez LM, Yang J, Smolka SA, Grosu R (2019) Statistical Model Checking. *Computing and Software Science: State of the Art and Perspectives* (Springer, Cham, Switzerland), , pp 478–504. https://doi.org/10.1007/978-3-319-91908-9_23

[2] Hérault T, Lassaigne R, Magniette F, Peyronnet S (2004) Approximate Probabilistic Model Checking. *International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI'04*, , pp 73–84.

[3] Younes HLS (2005) *Verification and Planning for Stochastic Processes with Asynchronous Events*. Ph.D. thesis. Carnegie Mellon, .

[4] The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

[5] A link to the repository will be provided soon.

[4] Filliben JJ (2003) *Lag Plot* (National Institute of Standards and Technology), . Available at https://www.itl.nist.gov/div898/handbook/eda/section3/lagplot.htm.
[5] Filliben JJ (2003) *Maximum Likelihood* (National Institute of Standards and Technology), . Available at https://www.itl.nist.gov/div898/handbook/eda/section3/eda3652.htm.
[6] Dataplot homepage. Available at https://www.itl.nist.gov/div898/software/dataplot/homepage.htm.
[7] Delignette-Muller M, Dutang C (2015) fitdistrplus: An R Package for Fitting Distributions. *Journal of Statistical Software* 64(4):1–34.
[8] Schittkowski K (2002) EASY-FIT: a software system for data fitting in dynamical systems. *Structural and Multidisciplinary Optimization* 23:153–169.
[9] Khoussi S, Heckert A, battou A, Bensalem S (2021) Neural networks for classifying probability distributions. *NIST* https://doi.org/10.6028/NIST.TN.2152. Available at https://nvlpubs.nist.gov/nistpubs/TechnicalNotes/NIST.TN.2152.pdf
[10] Filliben JJ (2003) *4-Plot* (National Institute of Standards and Technology), . Available at https://www.itl.nist.gov/div898/handbook/eda/section3/4plot.htm.
[11] Filliben JJ (1975) The Probability Plot Correlation Coefficient Test for Normality .
[12] Khoussi S (2021) Some real data for testing. *GitHub repository* .
[13] Dumonceaux R, Antle CE, Haas G (1973) Likelihood Ratio Test for discrimination between two models with unknown scale and location parameters. *Technometrics* 15(1):19.
[14] Dumonceaux R, Antle CE (1973) Discrimination Between the Log-Normal and the Weibull Distributions. *Technometrics* 15(4):923–926.
[15] Sherstinsky A (2020) Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D* 404:132306. https://doi.org/10.1016/j.physd.2019.132306
[16] Staudemeyer RC, Morris ER (2019) Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv* 1909.09586 Available at https://arxiv.org/abs/1909.09586v1.