Measuring and Comparing the Performance of Evolutionary Algorithms for Automatic Machine Learning for Trend Prediction in Time Series Data

Adam Lewison
lwsada002@myuct.ac.za
University Of Cape Town
Cape Town, South Africa

James Taljard
tljjam001@myuct.ac.za
University Of Cape Town
Cape Town, South Africa

KEYWORDS

Automated Machine Learning, Deep Neural Networks, Time Series Forecasting, Trend Prediction

1 PROJECT DESCRIPTION

In the past few decades, machine learning techniques have attracted a lot of attention from both the academic and commercial industries due to their proven success in various application domains [6]. However, developing a useful machine learning (ML) model is a repetitious, complex and time consuming process that must usually be undertaken by a machine learning expert. The ML expert is required at all stages of the machine learning pipeline in order to ensure that the model will be optimal. Automated Machine Learning (AutoML) is concerned with automating these stages of the ML pipeline. The automation of these different components reduces the human effort necessary for applying machine learning, improves the performance of machine learning algorithms, and improves the reproducibility and fairness of scientific studies [5].

Automatic Machine Learning (AutoML) has been evolving for a number of decades. As early as the 1990s, solutions were available for hyperparameter optimisation for selected classification algorithms using the brute-force [15] grid search technique. In 2011, Bayesian Optimisation was shown to be the state-of-the-art AutoML methodology for Hyper-Parameter optimisation [2]. In 2013 a open-source software package was introduced [18] in order to tackle the challenge of automating the process of Combined Algorithm Selection and Hyperparameter optimisation (CASH). This problem is commonly formulated as an optimization problem that can be solved by a wide range of techniques [6].

In time series trend prediction we are interested in understanding and forecasting the evolving trend of a time series i.e. the upward or downward pattern. According to [19] in certain cases it is more valuable to predict the trend of a time series than to forecast a future point value of the time series.

In recent years, researchers [12, 19] have used AutoML methodology in order to create future predictions about trend for given time series data. Kouassi and Moodley made a break through in the research in applying AutoML techniques specifically to time series trend prediction and obtained state-of-the-art results. However, in their research, the only optimisation technique that was used was a modified version of Bayesian optimisation, BOHB.

In this project, we will focus on the use and performance of evolutionary algorithms (genetic algorithms and particle swarm optimisation) in comparison to other optimisation algorithms, namely random search, grid search, and Bayesian optimisation.

2 PROBLEM STATEMENT

In our research we focus on a subset of the Automated Machine Learning pipeline which is concerned with finding the optimal combination of machine learning algorithm and related hyperparameters. As such, we explore the performance of evolutionary algorithms to solve the black-box CASH optimisation problem. The domain in which we solve this problem is that of trend prediction in time series data. Therefore we use genetic algorithms and particle swarm optimisation in order to select an appropriate ML algorithm and its optimal hyperparameters.

2.1 Research Objectives

In order to tackle this problem, our research has the following aims:

(1) Analyse and compare the performance of the following optimisation techniques when applied to the CASH problem in the context of time series trend prediction
  (a) Grid Search
  (b) Random Search
  (c) Bayesian Optimisation
  (d) Particle Swarm Optimisation
  (e) Genetic Algorithms

2.2 Research Questions

Upon completing the above objectives, we aim to be able to answer the following questions:

(1) Do Evolutionary Algorithms perform better as a CASH solver than the brute-force algorithm (grid-search) when used for time series trend prediction?
(2) Do Evolutionary Algorithms perform better as a CASH solver than other state-of-the-art optimisation techniques when used for time series trend prediction?
(3) Which of Particle Swarm Optimisation and Genetic algorithms performs better as a CASH solver when used for time series trend prediction?

3 PROCEDURES AND METHODS

3.1 CASH Search Space

The CASH search space can be divided into two fundamental parts: the algorithms under consideration and their respective hyperparameters. These algorithms are a multi-layer perceptron (MLP), convolutional neural network (CNN), and a long short-term memory (LSTM). Most of the hyperparameters are common to all of these above mentioned ML algorithms, while some algorithms have

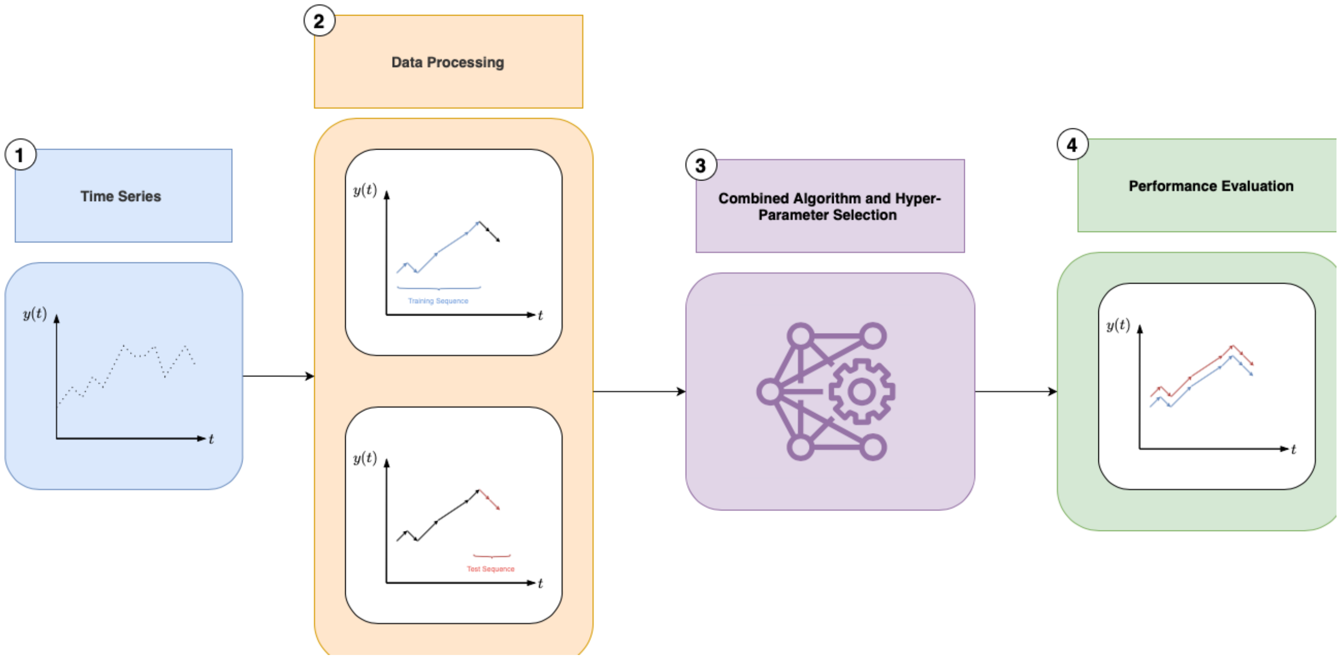

Figure 1: An overview of the procedure of using AutoML for time series trend prediction

hyperparameters that are specific to them. Common hyperparameters include number of hidden layers, number of hidden neurons in layer i, and learning rate. The LSTM has an additional hyperparameter - batch size. A CNN has three additional hyperparameters: pooling type for CNN layers, pooling size for CNN layers, and the kernel size for CNN layer i.

3.2 Data Sets

Experimentation will be perform using four time series data sets. The first three data sets are the daily closing stock prices of three financial stocks/indices. The fourth data set is a time series of household power consumption in KW.

3.2.1 NYSE Daily Closing Price. The first data set is the daily closing prices of the NYSE from 31/12/1965 to 22/06/2021. The data set was obtained from Yahoo Finance and contains 13964 data points.

3.2.2 Satrix 40 Daily Closing Price. The second data set is the daily closing prices of the Satrix 40 stock index from 02/01/2008 to 22/01/2021. The data set was obtained from Yahoo Finance and contains 3454 data points.

3.2.3 Nasdaq Daily Closing Price. The third data set is the daily closing prices of the Nasdaq stock from 05/02/1971 to 22/06/2021. The data set was obtained from Yahoo Finance and contains 12706 data points.

3.2.4 Household Power Consumption in KW. The fourth data set is the household power consumption in KW measured every minute. The data set was obtained from [1] and contains 2075259 data points.

[1]https://archive.ics.uci.edu/ml/machine-learning-databases/00235/

3.3 Data Processing

As depicted in step one and two of Figure 1. The input into our procedure is raw time series data and is obtained from the data sets mentioned in 3.2. A time series is represented as a sequence X such that

$$X = [x_1, ..., X_T]$$

where x_t is a real-valued observation at time t. We aim to convert raw time series data X into a trend sequence T as in [12] and is obtained by performing a piece-wise linear approximation of X. In [10] a novel and efficient method is proposed to produce such sequence T. More precisely

$$T = [< \ell_1, s_1 >, ..., < \ell_k, s_k >]$$

where ℓ_i is the number of data points x_t covered by trend i and s_i is the slope of trend k.

3.4 Experiment Design

In order to answer our research questions identified in Section 2.2; we set out to conduct our experimentation on different optimisation techniques to solve the CASH problem. This can be done with a full grid experiment (depicted in Table 1), i.e, we apply each of the following optimisation techniques to each data set

(1) Grid Search
(2) Random Search
(3) Bayesian Optimisation
(4) Particle Swarm Optimisation
(5) Genetic Algorithm

Optimisation Technique	Data Set			
	1	2	3	4
Grid Search	Y_{11}	Y_{12}	Y_{13}	Y_{14}
Random Search	Y_{21}	Y_{22}	Y_{23}	Y_{24}
Bayesian Optimisation	Y_{31}	Y_{32}	Y_{33}	Y_{34}
Particle Swarm Optimisation	Y_{41}	Y_{42}	Y_{43}	Y_{44}
Genetic Algorithm	Y_{51}	Y_{52}	Y_{53}	Y_{54}

Table 1: An overview of the experimental design. Where each Y_{ij} is repeated ten times to account for random initialisation. The best performing model in each experiment $\mathbf{Y}_{ij} willalsobetestedtentimes.$

Grid search is considered a brute force solution [15] in which all combinations of the input space are evaluated and thus can be used as a standard benchmark for comparison. Random Search has been shown to have better performance over grid search and thus can be used as a second benchmark. Bayesian Optimisation has in many cases [12, 13, 18] proved to be a state-of-the-art optimisation technique in Combined Algorithm and Hyper-Parameter Selection [18] and Hyper-Parameter Optimisation [12] and is thus worthwhile to include in our experiment in order to test against the two evolutionary algorithms (Particle Swarm Optimisation and Genetic Algorithm) which is the subject of this research.

3.4.1 Replication. Each optimisation technique will be applied independently ten times to each data set, i.e., each Y_{ij} in Table 1 is a vector of experiments of size ten. This ensures that we can separate difference in resulting from randomised initialisation of deep learning algorithms from differences that result due to the difference in optimisation technique.

3.5 Optimisation and Parameter Settings

3.5.1 Evaluation. In order to evaluated our models, we will use walk-forward validation. This ensures that the order of data is preserved. Data will be roughly divided in the following proportions: Training Set (70

3.5.2 Objective Function. A primary goal when training a machine learning model is to minimise a specified loss function which indicates goodness of fit of predicted values to with the corresponding training values. To this end we employ a mean squared error loss function.

3.6 Performance Evaluation

3.6.1 Root mean square error. The evaluation on Trend Prediction used in [12] is Root mean square error (RMSE). Since errors are squared before taking the average, RMSE gives a higher weighting to large errors. Hence, RMSE is a useful metric when large errors are undesired. In our experimentation we define three RMSE statistics as follows

(1) Length Root Mean Square Error

$$= \sqrt{\frac{1}{T}\sum_{i=1}^{T}(\ell_i - \hat{\ell}_i)^2}$$

(2) Slope Root Mean Square Error

$$= \sqrt{\frac{1}{T}\sum_{i=1}^{T}(s_i - \hat{s}_i)^2}$$

(3) Combined Root Mean Square Error

$$= \sqrt{\frac{1}{T}\sum_{i=1}^{T}(\beta_1(\ell_i - \hat{\ell}_i) + \beta_2(s_i - \hat{s}_i))^2}$$

for some weights β_1 and β_2

3.7 Budget

We will work off of the work of [12], who experimentally found a reasonable upper bound for the *number of training epochs* required to identify optimal configurations.

3.8 Software

A discussion of the following choices

3.8.1 Automated Machine Learning Libraries.

DEAP. Introduced in 2012 in [8], *DEAP* is a powerful, yet simple, evolutionary computation package for Python that contains several evolutionary algorithms such as GA and PSO. And thus is a suitable candidate as a package to implement our AutoML tasks.

PSPSO. In recent years, Haidar et al. developed a “high-level” python package called *PSPSO* [9] which is used for selecting machine learning algorithms and respective hyper-parameters by using the particle swarm optimization algorithm. An advantageous quality of this package is the ability to manually add deep learning algorithms to the search space and thus can be tailored to our needs.

3.8.2 Machine Learning Library.

PyTorch. In 2019 Paszke et al. [16] introduced *Pytorch.*

PyTorch offers a user-friendly interface with a large variety of deep learning algorithms and as such has become an accepted tool in the deep learning research community. This is an ideal Python library to use to implement machine learning algorithms in our experimentation.

3.9 Computing Environment

3.9.1 Google Colab. Colaboratory by Google [2] is an ideal environment in order to run our experiments due to the fact that they offer access to a GPU’s and allow user installations.

4 ETHICAL, PROFESSIONAL, AND LEGAL ISSUES

4.1 Ethical Issues

There are no foreseeable ethical issues that may arise throughout the duration of the project.

4.2 Professional Issues

There are no prevalent professional issues are of concern to the team. Proper use of computing resources like the UCT HPC.

[2]https://colab.research.google.com

4.3 Legal Issues

There are no legal issues that may arise during the course of the project. The data sets have been obtained from Yahoo Finance and a machine learning repository and can be used for the purpose of research. The software tools that have been outlined in section 3.8 are all open source.

5 RELATED WORK

5.1 AutoML for Trend Prediction in Time Series Data

Literature regarding the use of AutoML for trend prediction in time series data is limited. Kouassi and Moodley [12] showed that the application of AutoML for HPO for deep learning algorithms can perform effectively when compared to deep learning algorithms with hand-tuned hyperparameters. The bulk of the literature dealt with the use of AutoML for HPO for machine learning models for time series forecasting. AutoML was applied to various machine learning models including long short-term memory, recurrent neural network, support vector regression, multilayer perceptrons, and gated recurrent units. Of these machine learning algorithms, LSTM and GRU appeared to be most applicable to time series data [1, 3, 11, 14]. However, most of the research focused on the performance of AutoML and a particular machine learning algorithm applied to time-series forecasting and not the performance of several AutoML techniques when applied to various machine learning algorithms when applied to time series data.

5.2 Genetic Algorithms for AutoML

Genetic algorithms are a subclass of evolutionary algorithms that draw inspiration from the process of natural selection. Due to there ability to solve non-convex, non-continuous, non-smooth optimisation problems, genetic algorithms have been effectively applied for hyperparameter optimisation, and AutoML in general. The majority of literature reviewed focused on the use of GAs for HPO for a particular ML algorithm as opposed to the larger CASH problem. In almost all cases, a hybrid GA model outperformed models with hand-tuned hyperparameters [3, 11].

5.3 Particle Swarm Optimisation for AutoML

Escalante et al. [7] formulated the *full model selection* problem, that consists of finding the best combination of data pre-processing, feature selection/extraction and classification models, along with hyper-parameter optimisation and Particle Swarm Optimisation was used to solve the problem.

Particle Swarm Optimisation used for Hyper-Parameter Optimisation [17]

In *Optunity*, a Software Package released in 2014 [4], Particle Swarm Optimisation was chosen as the default optimisation solver as their experiments proved Particle Swarm Optimisation to perform well for a large variety of tuning tasks involving various learning algorithms.

6 ANTICIPATED OUTCOMES

The primary focus of the research is to determine how evolutionary algorithms perform when applied to AutoML techniques, like HPO, compared to more standard techniques like grid search or Bayesian optimisation. As such, our results are anticipated to provide a meaningful way to compare these various AutoML techniques in order to determine if evolutionary algorithms are useful in the application to AutoML. Due to the nature of the CASH search space, we anticipate that these evolutionary algorithms will perform as well as, if not better than, the benchmark techniques. This is due to the fact that these evolutionary algorithms are well suited for complex solution spaces. The secondary focus of the research is to evaluate how well AutoML can be used in application to trend prediction in time series data. We anticipate that our models with perform adequately in terms of prediction accuracy. A meaningful comparison must be drawn in order to evaluate this point since we do not include any benchmark experiments that do not use AutoML to select the model hyperparameters.

7 PROJECT PLAN

7.1 Risks

Refer to appendix A for the risk matrix regarding the project.

7.2 Timeline

Refer to appendix B for a detailed Gantt chart that details the project timeline as well as work allocation and significant milestones.

7.3 Resources Required

Primary to the execution of AutoML is computing power, and thus necessary high performance computing services (as introduced in Section 3.9) is a key consideration. Moreover, proper planning for time will need to be considered because even with the use of high performance computing services certain optimisation will still take a number of hours to complete.

7.4 Deliverables and Milestones

The deliverables and milestones are shown in the Gantt chart in appendix B. These are illustrated by the yellow diamonds on the chart.

7.5 Work Allocation

The Gantt chart found in appendix B illustrates the distribution of the work amongst team members. The team will work together to prepare the data, run the benchmark tests, and compile the final report. Adam Lewison will solely be responsible for the tests regarding the use of PSO for AutoML, while James Taljard will be solely responsible for the tests regarding the use of GAs for AutoML.

ACKNOWLEDGMENTS

The authors would like to thank Dr Deshendran Moodley for his insight and guidance.

REFERENCES

[1] Abdulaziz Almalaq and Jun Jason Zhang. 2019. Evolutionary Deep Learning-Based Energy Consumption Prediction for Buildings. *IEEE Access* 7 (2019). https://doi.org/10.1109/ACCESS.2018.2887023
[2] James Bergstra, R Bardenet, Yoshua Bengio, Balázs Kégl, and Rémi Bardenet. 2011. Algorithms for Hyper-Parameter Optimization. https://hal.inria.fr/hal-00642998

[3] Salah Bouktif, Ali Fiaz, Ali Ouni, and Mohamed Adel Serhani. 2018. Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies* 11, 7 (2018). https://doi.org/10.3390/en11071636
[4] Marc Claesen, Jaak Simm, Dusan Popovic, Yves Moreau, and Bart De Moor. 2014. Easy Hyperparameter Search Using Optunity. arXiv:1412.1114 [cs.LG]
[5] Frank. editor. Hutter, Lars. editor. Kotthoff, and Joaquin. editor. Vanschoren. 2019. *Automated Machine Learning*. Springer International Publishing. 101822 pages. https://doi.org/10.1007/978-3-030-05318-5
[6] Radwa Elshawi, Mohamed Maher, and Sherif Sakr. 2019. Automated Machine Learning: State-of-The-Art and Open Challenges. (6 2019). http://arxiv.org/abs/1906.02287
[7] Hugo Jair Escalante, Manuel Montes, and Luis Villaseñor. 2009. Particle Swarm Model Selection for Authorship Verification. https://doi.org/10.1007/978-3-642-10268-4_66
[8] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary algorithms made easy. *The Journal of Machine Learning Research* 13, 1 (2012), 2171–2175.
[9] Ali Haidar, Matthew Field, Jonathan Sykes, Martin Carolan, and Lois Holloway. 2021. PSPSO: A package for parameters selection using particle swarm optimization. *SoftwareX* 15 (2021), 100706.
[10] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. 2001. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE international conference on data mining*. IEEE, 289–296.
[11] Trang Thi Kieu Tran, Taesam Lee, Ju Young Shin, Jong Suk Kim, and Mohamad Kamruzzaman. 2020. Deep learning-based maximum temperature forecasting assisted with meta-learning for hyperparameter optimization. *Atmosphere* 11, 5 (2020), 1–21. https://doi.org/10.3390/ATMOS11050487
[12] Kouame Hermann Kouassi and Deshendran Moodley. 2020. Automatic deep learning for trend prediction in time series data. (9 2020). http://arxiv.org/abs/2009.08510
[13] Tao Lin, Tian Guo, and Karl Aberer. 2017. Hybrid Neural Networks for Learning the Trend in Time Series. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2273–2279. https://doi.org/10.24963/ijcai.2017/316
[14] Jiseong Noh, Hyun Ji Park, Jong Soo Kim, and Seung June Hwang. 2020. Gated recurrent unit with genetic algorithm for product demand forecasting in supply chain management. *Mathematics* 8, 4 (2020). https://doi.org/10.3390/math8040565
[15] Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. 2016. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. arXiv:1603.06212 [cs.NE]
[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]
[17] Quan Sun, Bernhard Pfahringer, and Michael Mayo. 2012. Full model selection in the space of data mining operators. (07 2012). https://doi.org/10.1145/2330784.2331014
[18] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 2013. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. https://doi.org/10.1145/2487575.2487629
[19] Min Wen, Ping Li, Lingfei Zhang, and Yan Chen. 2019. Stock Market Trend Prediction Using High-Order Information of Time Series. *IEEE Access* 7 (2019). https://doi.org/10.1109/ACCESS.2019.2901842

A RISK MATRIX

Risk Condition:	Impact:	Probability:	Mitigation/Management:
Team member contracts COVID	Moderate	Low	Planning well and completing milestones as soon as possible. Team members need to ensure they stay safe and avoid contracting the virus.
Time series data is not useful for the study.	High	Low	Ensure that the data is sufficient in size and there are trends present in the data. Useful datasets have already been identified.
Poor communication amongst team members.	Moderate	Low	Team member will know what their responsibilities are and must ensure that deadlines are met on time. Regular meetings will be held in order to ensure members are on the same page and work will be done together where possible.
Time to run tests is too great.	High	Moderate	Selecting the budget for each test must be carefully selected in order to ensure that test do not take too long. Beginning testing early and carefully planning the experiments will allow for additional time in the event that some tests take too much time.
Difficulty integrating DEAP with PyTorch.	High	Low	The team must be flexible in order to be able to change the software environment to suit the project needs. As DEAP is a common python package for evolutionary algorithms, it can integrate well with PyTorch and should not present a problem.
Results of experiments do not support the problem questions	High	Moderate	Design of experiments must be carefully considered to ensure that research questions are answered. Appropriate evaluation metrics must be selected in order to effectively analyse the results.
Poor time management	Moderate	Moderate	The team must follow the Gantt chart carefully, but also be flexible enough to accommodate any changes to the project plan.

B GANTT CHART

teamgantt
Created with Free Edition