Building a Time-Series Forecast Model with Automated Machine Learning for Heart Rate Forecasting Problem

Huu-Anh-Duc Cap*,†, Trong-Hop Do*,†, Demeke Shumeye Lakew‡, and Sungrae Cho‡

* University of Information Technology, Ho Chi Minh City, Vietnam.
† Vietnam National University, Ho Chi Minh City, Vietnam.
‡ School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea.

***Abstract*—Time series forecasting is currently a very popular field of study. Easily find a variety of time series data in medicine, weather forecasting, biology, supply chain management, stock price forecasting, and more. With the proliferation of data and computing power in recent years, deep learning has become the first choice for building time series predictive models. While traditional Machine Learning models – such as autoregression (AR), Exponential smoothing, or autoregressive integrated moving average (ARIMA) – perform manual conversion of the original raw data set into a set of attributes, and the optimization of the parameter must also be based on feature selection, the Deep Learning model only learns the features directly from the data alone. As a result, it speeds up the data preparation process and can fully learn more complex data patterns. In this paper, we designed LSTM deep learning network using Automated Machine Learning (AutoML) method to predict time series data which is the heart rate data. The results of this model can be applied to the field of medicine and health care.**

***Index Terms*—Hear rate, time series forecasting, automated machine learning**

I. Introduction

Heart rate is one of the vital signs of the body's health as measured by the number of times the heart contracts (heart beats) in a minute. The heart rate in a normal person can vary depending on whether the body is active or at rest, the stress level of the nervous system, changes in health, age, or some other impact. The heart rate in a normal human is cyclical, and regular, and in a human lifetime, the heart beats about 3000 million times. Each heart cycle works independently, individually, and is measured as the interval from the beginning of one heart sound to the beginning of another. Classically, the interval between the first and second heart sounds is called a heartbeat. Although heart rate can't tell the body's health status, it is the standard in diagnosis as well as the earliest and fastest sign to identify possible abnormalities. For this reason, predicting heart rate changes in advance is of great significance. In this paper, we will apply deep learning technology to predict heart rate data. The problem of analysis and prediction of time series has been studied extensively for more than 40 years. Given the time series t1, t2, ..., tn, ..., we need to estimate the value at time i based on the previous data. Some of the traditional machine learning models used for time series forecasting include ARIMA and Exponential smoothing. ARIMA is a combination of Autoregression (AR) and Moving Average (MA) approaches in building synthetic models of time series. This model is quite simple but can give good results. It includes parameters to consider seasonality, long-term trends, autoregression and moving averages, from which autocorrelation is embedded in the data. However, the disadvantages of traditional Machine Learning are:

- Missing features can affect the performance of the models
- Machine learning models cannot recognize complex patterns in data
- Machine learning does not perform well in long-term forecasting

The use of deep learning for time series forecasting has overcome the limitations of traditional machine learning with various approaches. Currently, Regression Neural Network (RNN) is the classical and most used architecture for Time Series Forecasting problems. RNN has advantages in that:

- The performance of the RNN is not significantly affected by the missing values
- RNNs can find complex patterns in the input time series
- RNNs give good results in long-term forecasting, more than just a few steps
- RNNs can model data series to assume correlations between samples

However, some limitations of RNNs are:

- When trained on long time series, RNNs often encounter gradient vanishing/exploding problem. That means the parameters in the hidden layers do not change much or lead to metric and behavioral instability. This is because the gradient of the function affects its memorability.
- RNNs have limited memorization, so some elements of the past may be overlooked in predicting the future.
- RNN training is complex and expensive in terms of computational infrastructure.

Due to these disadvantages, various extensions of RNN have been designed to reduce the internal memory. The Long-Term Memory Model (LSTM) [1] shows good results with the time series problem being one of them. However, this success based completely on experienced professionals taking important steps in the process of creating a good machine

2022 13th International Conference on Information and Communication Technology Convergence (ICTC) | 978-1-6654-9939-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTC55196.2022.9952797

learning model. Automated Machine Learning [2] [3] rovides methods and processes to make Machine Learning available for non-Machine Learning experts [4], to improve efficiency of Machine Learning and to accelerate research on Machine Learning.

II. Background

A. Apache Spark

Apache Spark is a powerful and fast open-source cluster computing framework with an increasing number of use cases in the industry. The key features such as speed and scalability make it a great alternative to Hadoop. It is much faster than Hadoop, especially with batch processing, which allows it to deal with huge datasets in just a matter of a few minutes. The foremost reason why Apache Spark is ruling in the big data industry is its outstanding in-memory data processing. Most tasks of Apache Spark take place in in-memory. This makes it faster and more optimized as compared to other approaches like Hadoop's MapReduce. Apache Spark has outstanding potentials that can boost the big data-related businesses in the industry.

B. BigDL

BigDL is a distributed deep learning library for Apache Spark; with BigDL, users can write their deep learning applications as standard Spark programs, which can directly run on top of existing Spark or Hadoop clusters. BigDL makes it easy for data scientists and data engineers to build end-to-end, distributed AI applications. The BigDL 2.0 release combines the original BigDL and Analytics Zoo projects, providing the following features:

- DLlib: distributed deep learning library for Apache Spark
- Orca: seamlessly scale out TensorFlow and PyTorch pipelines for distributed Big Data
- RayOnSpark: run Ray programs directly on Big Data clusters
- Chronos: scalable time series analysis using AutoML
- PPML: privacy preserving big data analysis and machine learning (experimental)
- Nano: automatically accelerate TensorFlow and PyTorch pipelines by applying modern CPU optimizations

C. Chronos

Chronos is an application framework for building large-scale time series analysis applications. The Chronos architecture is shown in Fig. 1.

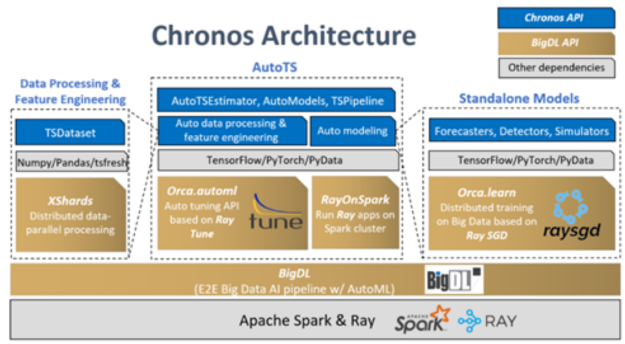

Fig. 1: Chronos Architecture.

D. LSTM Network

Long Short-Term Memory Network is an advanced RNN, a sequential network, that allows information to persist. It is capable of handling the vanishing gradient problem faced by RNN. A recurrent neural network is also known as RNN is used for persistent memory. It is applied very effectively in many problems and is now widely used. LSTM is designed to avoid the long-term dependency problem. Remembering information for long periods of time is the default behavior of the network. Regressive neural networks take the form of a repeating sequence of neural network modules. In RNN, these modules have a very simple structure, just a fishy layer. LSTMs also have a string structure, but the modules have a different structure. Instead of having a single neural network layer, LSTM has 4 layers (shown in Fig. 2), which interact with each other. The core idea of LSTM [5] is that the state

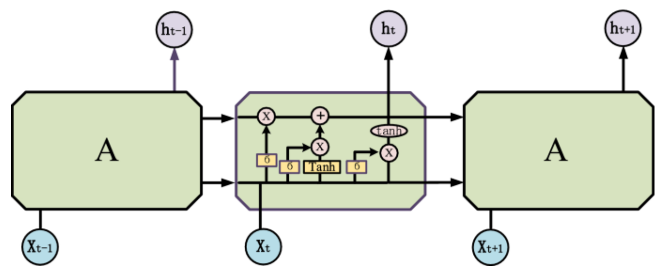

Fig. 2: LSTM Model.

of the cells is depicted by the horizontal line at the top. The cell state is like a carousel. It runs straight through the whole chain, with only a few small linear interactions, it is easy for information to flow along unchanged. LSTMs [6] can remove or add information to the cellular state, which is carefully regulated by structures called gates. The gate is an optional way for information to pass through. They are composed of a layer of sigmoid neural networks and a multiplication operator. The output of the sigmoid layer is the values in the interval [0, 1], describing the throughput of each component. A value of 0 means "nothing through", while a value of 1 means "let everything pass". An LSTM has three sigmoid gates to protect and control the cell state. However, building an LSTM network [7] architecture for a specific problem must go through complicated stages:

- Preprocess and clean the data.
- Select and construct appropriate features.
- Select an appropriate model family.
- Optimize model hyperparameters: Hyper-Parameter Optimization is most mature subarea in AutoML [8]
- Design the topology of neural networks (if deep learning is used): pooling or fully connection ...
- Postprocess machine learning models.

- Critically analyze the results obtained.

As the complexity of these tasks often exceeds that of non-ML experts, the rapid growth of machine learning applications has created the need for stand-alone machine learning methods that can be used Easy to use and requires no specialized knowledge This research area targets the progressive automation of machine learning - AutoML [9].

III. Experimental

A. Exploratory Data Analysis

The average heart rate data collected per minute from April 12 to May 12 with 45.000 records.Table. I will give some example records.

TABLE I: Training data

dt	Value
2016-04-12 00:00:00	64
2016-04-12 00:01:00	64
2016-04-12 00:02:00	64
2016-04-12 00:03:00	64
2016-04-12 00:04:00	64
2016-04-12 00:05:00	64
2016-04-12 00:06:00	65
2016-04-12 00:07:00	65

An early step in any effort to analyze or model data should be to understand how the variables are distributed. Techniques for distribution visualization can provide quick answers to many important questions. What range do the observations cover? What is their central tendency? Are they heavily skewed in one direction? Fig. 3 and Fig. 4 provide an overview of the data.

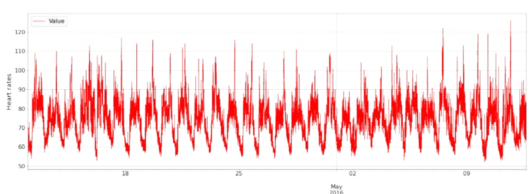

Fig. 3: Data visualizations.

A given time series is thought to consist of three systematic components including level, trend, seasonality, and one non-systematic component called noise.

These components are defined as follows:

- Level: The average value in the series.
- Trend: The increasing or decreasing value in the series.
- Seasonality: The repeating short-term cycle in the series.
- Noise: The random variation in the series.

Time series decomposition involves thinking of a series as a combination of level, trend, seasonality, and noise components. Decomposition (shown in Fig. 5) provides a useful abstract model for thinking about time series generally and for better understanding problems during time series analysis and forecasting.

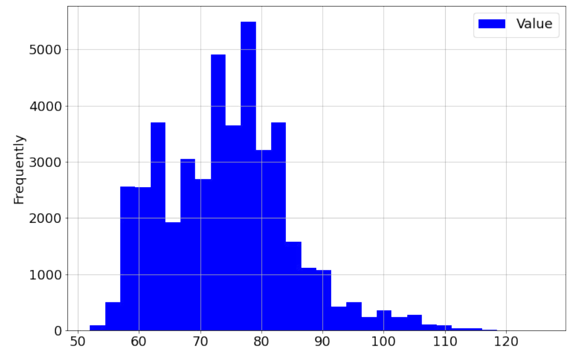

Fig. 4: Data distributions.

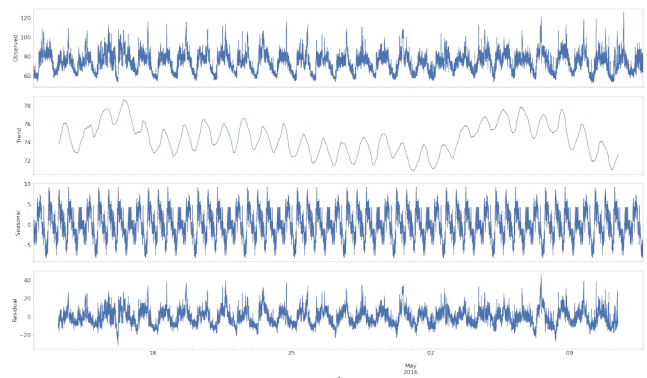

Fig. 5: Data decomposition analysis.

B. Automated Feature engineering using TS Dataset and Sklearn Standard Scaler

Time series data is a special formula of data with specific operations. TSDataset is an abstraction of a time series dataset, providing various data processing operations (e.g., display, deduplication, resampling, scale/unscale, scroll) and feature engineering methods (e.g., datetime feature, composite feature). Calling cascade is supported for most methods. TSDataset can be initialized from a pandas dataframe and converted to a pandas dataframe or a numpy ndarray. It will be used for the AutoML training process. Figure 6 is a sample in the training Tsdataset.

	dt	Value	id	MINUTE	DAY	DAYOFYEAR	HOUR	WEEKDAY	WEEKOFYEAR	MONTH	YEAR	IS_AWAKE	IS_BUSY_HOURS	IS_WEEKEND
0	4/12/2016 0:00	-1.29943	0	-1.70301	-0.56389	-1.661531	-1.64558	-0.996112	-1.344604	-0.54023	0	0.518988	-0.639538	-0.568801
1	4/12/2016 0:01	-1.00766	0	-1.64527	-0.56389	-1.661531	-1.64558	-0.996112	-1.344604	-0.54023	0	0.518988	-0.639538	-0.568801
2	4/12/2016 0:02	-1.00766	0	-1.58754	-0.56389	-1.661531	-1.64558	-0.996112	-1.344604	-0.54023	0	0.518988	-0.639538	-0.568801
3	4/12/2016 0:03	-1.00766	0	-1.5298	-0.56389	-1.661531	-1.64558	-0.996112	-1.344604	-0.54023	0	0.518988	-0.639538	-0.568801
4	4/12/2016 0:04	-1.00766	0	-1.47206	-0.56389	-1.661531	-1.64558	-0.996112	-1.344604	-0.54023	0	0.518988	-0.639538	-0.568801
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
35340	5/6/2016 13:00	-0.32688	0	-1.70301	-1.24053	1.724275	0.234448	0.576708	1.514044	1.85106	0	0.518988	-0.639538	-0.568801
35341	5/6/2016 13:01	-0.22962	0	-1.64527	-1.24053	1.724275	0.234448	0.576708	1.514044	1.85106	0	0.518988	-0.639538	-0.568801
35342	5/6/2016 13:02	-0.22962	0	-1.58754	-1.24053	1.724275	0.234448	0.576708	1.514044	1.85106	0	0.518988	-0.639538	-0.568801
35343	5/6/2016 13:03	0.062145	0	-1.5298	-1.24053	1.724275	0.234448	0.576708	1.514044	1.85106	0	0.518988	-0.639538	-0.568801
35344	5/6/2016 13:04	-0.03511	0	-1.47206	-1.24053	1.724275	0.234448	0.576708	1.514044	1.85106	0	0.518988	-0.639538	-0.568801

Fig. 6: Datetime features.

C. Build model with AutoTS

AutoTS provides AutoML support for building end-to-end time series analysis pipelines (including automatic feature generation, model selection and hyperparameter tuning). The general workflow using automated training contains below two steps:

- Create a AutoTSEstimator to train a TSPipeline, save it to file to use later or elsewhere if you wish.
- Use TSPipeline to do prediction, evaluation, and incremental fitting as well.

D. Mean Square Error (MSE)

The project use MSE as loss function. The MSE either assesses the quality of a predictor (i.e., a function mapping arbitrary inputs to a sample of values of some random variable), or of an estimator (i.e., a mathematical function mapping a sample of data to an estimate of a parameter of the population from which the data is sampled). Equation 1 provides the definition of the MSE:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{1}$$

where N is the number of samples, y_i and $\hat{y}_i$ are the exact and predicted results, respectively.

E. Results

The MSE value of which achieved with the testing dataset is 9.4. This is almost similar to the performance achieved with an LSTM model built manually.

The configurations of the implemented LSTM model (parameters, hyperparameter, ...) are listed in Table II.

TABLE II: Model configuration

Attribute	Value
batch_size	32
dropout	0.17876954051566896
future_seq_len	1
hidden_dim	32
input_feature_num	9
layer_num	2
lr	0.0027737364469418873
output_feature_num	1
past_seq_len	5
selected_features	'DAYOFYEAR','WEEKDAY', 'WEEKOFYEAR', 'MONTH', 'IS_BUSY_HOURS', 'YEAR', 'HOUR','IS_AWAKE'

The predicted and actual values is shown in Fig. 7.

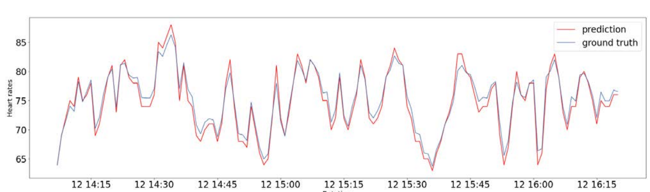

Fig. 7: Results based on test dataset for about 2 hours.

IV. Conclusion

With the proliferation of data and computing power in recent years, deep learning has become the first choice for building time series predictive models. Using Automated Machine Learning (AutoML) methods to build LSTM deep learning networks is highly effective for heart rate prediction problems, especially when deep learning network implementers are not experts. The results of this model can not only be applied to the field of medicine and health care but also open a new direction in building and deploying deep learning network models for many other problems in the field of reality.

Acknowledgment

This work was supported by the Korea Electric Power Corporation under Grant R20XO02-15.

References

[1] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[2] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

[3] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[4] J. Waring, C. Lindvall, and R. Umeton, "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare," *Artificial intelligence in medicine*, vol. 104, p. 101822, 2020.

[5] D.-H. Shin, K.-H. Choi, and C.-B. Kim, "Deep learning model for prediction rate improvement of stock price using rnn and lstm," *The Journal of Korean Institute of Information Technology*, vol. 15, no. 10, pp. 9–16, 2017.

[6] S. Anveshrithaa and K. Lavanya, "Real-time vehicle traffic analysis using long short term memory networks in apache spark," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. IEEE, 2020, pp. 1–5.

[7] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE, 2017, pp. 1643–1647.

[8] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[9] E. LeDell and S. Poirier, "H2o automl: Scalable automatic machine learning," in *Proceedings of the AutoML Workshop at ICML*, vol. 2020, 2020.