Time Series Feature Extraction and Selection for Fire Data

Wai Cheong Tam, Jun Wang
Fire Research Division, National Institute of Standards and Technology, Gaithersburg, MD, USA

Youwei Jia
Department of Electrical and Electronic Engineering, Southern University of Science and Technology, Guangdong, China

Eugene Yujun Fu
Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Jiajia Li
Department of Industrial Design, Guangdong University of Technology, Guangdong, China

Abstract

This paper aims to facilitate the use of machine learning to carry out supervised classification/regression tasks for time series data in fire research. Specifically, a feature engineering tool, FAST (Feature extrAction and Selection for Time-series), is developed. Using hypothesis test method together with principal component analysis, relevant features with high significance to the prediction are selected. A study case is presented for the use of FAST. Results demonstrate the importance of obtaining effective features and its potential benefits. It is expected that the feature engineering tool can help scientists and engineers in the fire research community develop accurate machine learning based models.

Keywords: Time series; detection; feature extraction; feature selection; machine learning

Introduction

Machine learning (ML) paradigm is among the top methods that can be used to overcome real-time detection in many disciplines. Cao and his co-worker [1] provided a support vector machine being trained based on historical financial data for unusual trading detection in stock markets. Using streaming signals from vibration sensors in multiple locations, Anton et al. [2] developed a neural network-based model to detect potential intrusion on an industrial machine. It was reported that the model can provide early warning to schedule maintenance thus avoiding mechanical failure for large scale, high cost equipment.

Recently, research efforts are made to implement ML paradigms to overcome detection problems that traditional approaches such as physics-based models [3] might not be able to handle in the fire research community. Although the studies [4] showed that the use of ML paradigm provides improvement in detection performance with significant reduction to nuisance alarms, two technical difficulties were identified.

Foremost, relevant data is limited. In fire research community, it can be easily shown that acquiring the desired data, such as temperature, heat flux, velocity, and species concentration, is not trivial because a) the event of interest (i.e. fire ignition in a compartment) does not happen frequently, b) time series data associated with fire event are not available to the public data warehouse [5], and c) physically conducting fire experiments is costly and time-consuming. Although fire simulation programs (i.e. Fire Dynamic Simulator [3]) can be used to generate synthetic data for ML task, the physical models being used in these codes are simplified and will not capable the correct physics of fundamental processes such as gasification and pyrolysis. For that, mutual collaboration between laboratories might have to be established to resolve the data problem.

In additional to problems associated with data availability, another technical difficulty for the use of ML paradigms in fire research community is that the data is complex. Figures 1 show two sets of time series data from the cooktop ignition study reported in [6]. Three selected measurements are shown: velocity, carbon monoxide (CO), and volatile organic compounds (VOCs). Since the measurements are made with different devices, each of the time series is being normalized to its peak value obtained for a test for illustration purpose. As it is shown in the figures, different profiles are observed for the VOCs data. While the data associated with oil has a monotonic increasing trend, the food data has multiple peaks.

Also, data such as VOCs and CO (refer to left figure) have unphysical negative values with a number of missing data. More than that, the value associated with CO has a rapid increasing trend and becoming significant towards the end of a test. For velocity data, they fluctuate around a mean value. Lastly, it is worth noting that these measurements (time series) are rather non-stationary and this implies that the absolute values of these measurements are sensitive to the surroundings. These complexities make the application of the use of the ML paradigm difficult.

In order to train an accurate ML model, it is well understood that features[1] are required. Typically, feature engineering is being carried out to obtain effective features and this process requires an exhaustive amount of

[1] Features can be considered as the transformed data that better represents the underlying problem to the predictive models which helps to improve the model accuracy on unseen data.

work. Although one can use the existing ML packages (i.e. tsfresh [7] and tslearn [8]), the obtained features are often irrelevant and difficult to interpret. Consequently, none of these ML packages are being used in fire research community. In this paper, a feature engineering tool, FAST (Feature extrAction and Selection for Time-series), is introduced to facilitate the implementation of ML paradigm in fire research. In order to demonstrate the effectiveness of the use of FAST for development of a ML model, a case study is provided.

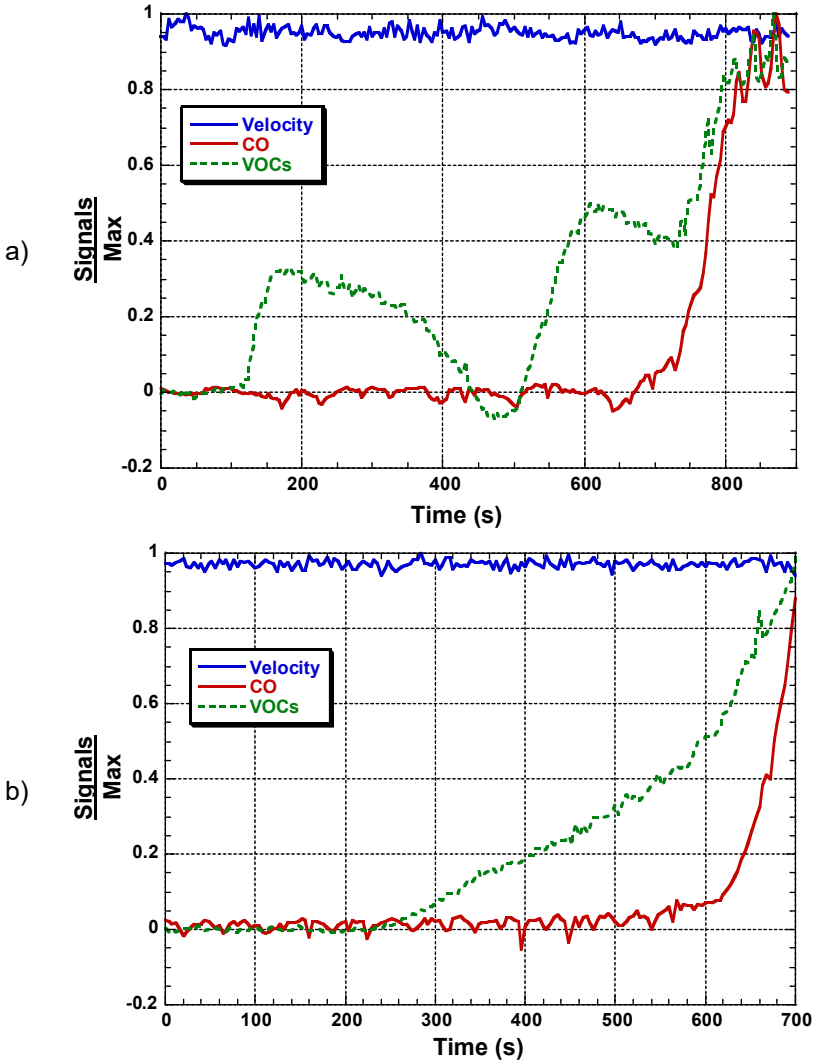

Fig. 1. Time series data examples: a) measurements for a food and b) measurements for oil.

Development of FAST

The automated feature engineering tool, FAST, consists of two modules: 1) feature extraction and 2) feature selection. Figure 2 shows the overall structure of FAST and its operational procedures. In the following section, the details of each module and the descriptions of FAST operational procedures are presented. It should be noted that more effective features help to provide simpler models, better results, and higher numerical efficiency.

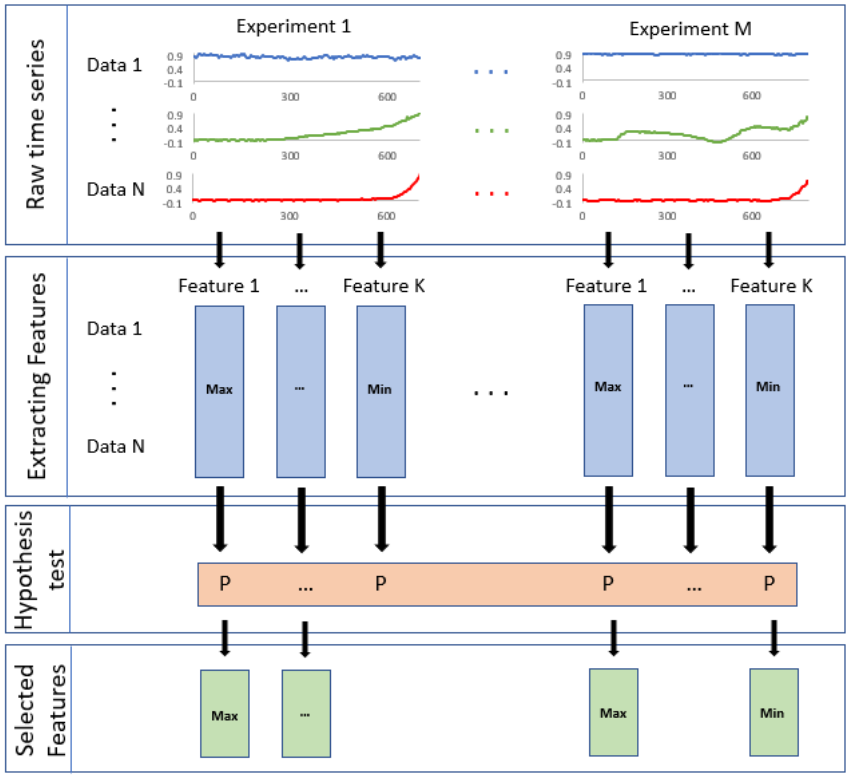

Fig. 2. Overall structure of FAST and its operation procedures.

Feature extraction module (1 – 2)

In the current development of FAST, time series data, such as kitchen fire [6], smart firefighting technology [9], and wildland fire [10], are being considered. The behavior of these data varies significantly in which they can be either monotonic, periodic, or oscillatory. In order to capture the important tendency and/or pattern of these data, four types of feature extraction methods are being utilized and based on i) statistics, ii) trends, iii) its correlation, and iv) domain (i.e. frequency) of interest.

In general, the application of statisticaly-based features such as mean or median over a sliding window[2] would help to obtain an average value and reduce the effect of data fluctuation due to outliers.

For trend-based features, they provide information about the data with respect to time. In many regression problems, one might be interested in providing forecast of a future event and correlation-based features offer flexibility to obtain highly non-linear information. The last type of feature (frequency domain based) capture crucial information in frequency domain that is difficult to deduce in time domain.

As shown in Fig. 2, the raw data will be separated into individual time series data in procedure 1. The data is then fed into the feature extraction module and a set of features (i.e. Min, Max, etc. as shown in procedure 2 in Fig. 2) are obtained. Table 1 provides a list of features that can be obtained in the feature extraction module. It should be noted that since the module is written to perform the extraction procedure separately, additional features can be added if necessary.

Table 1. List of features being included in FAST.

Statistical based	Min	Max	Mean
	Median	Standard deviation	Sum
	Count above mean	Count below mean	Number of peaks
	Number of crossing		
Trend based	Delta	Rate of change	Acceleration
Correlation based	1st order regression	Max slope	Sinusoid
	2nd order regression	Exponential function	
Frequency domain based	Main frequency	Secondary frequency	
	Coefficients for discrete Fourier transform		

Feature filtering module (3 – 4)

Two operations are carried out in the feature filtering module using hypothesis test and principal component analysis. The best 10 features are selected and they be readily used in training and testing of a ML algorithm.

[2] A sliding window contains a specified length of time series data (i.e.30 seconds) and it moves over the data, sample by sample, and features are extracted over the data in the window.

Given the experimental dataset obtained in [7], which is partially being illustrated in Fig. 1, more than a hundred of features are obtained from the feature extraction module. The number of features would be 5 to 10 times more if methods from [9,10] are used. In general, having this large number of features is not ideal because the model can easily be overfitted and/or is hard to interpret. For that, a hypothesis test [12] is used to filter out the irrelevant features. In [7, 10, 11], the extracted features are usually continuous and labels can either be continuous and binary.

For that, two test methods are adapted: i) Kolmogorov-Smirnov test [13] for cases where the features are continuous and the labels[3] are binary, and ii) Kendal rank test [14] for which both features and labels are continuous. Using either of the test methods, a p-value is obtained for each feature and the p-value quantifies the feature relevance to the prediction of the label. In principle, the smaller the p-value, the better/more relevant the features.

Fig. 3 shows the top 10 selected features extracted from the kitchen fire dataset mentioned in [7] ranked by its p-value. In addition, two of the irrelevant features (most right) are also included in the figure. It can be seen that the velocity data is not useful and it can be understood based on the data behavior shown in Fig. 1.

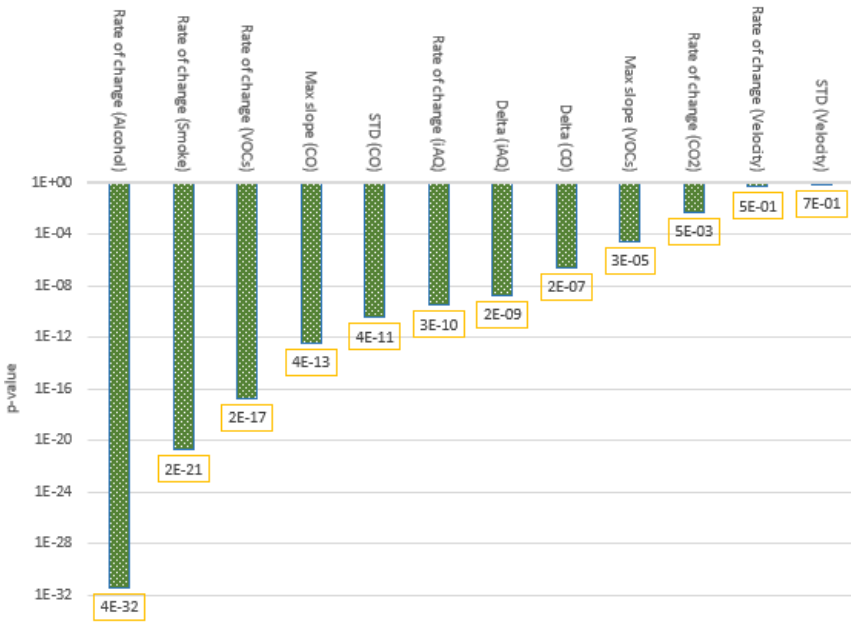

Fig. 3. Selected features with small p-value.

[3] Labels are referred to the targets for a classification/regression task.

Even though the hypothesis test helps to filter out a major portion of irrelevant features, there is still a high possibility that some of the selected features are redundant. For example, the features for median and mean are highly correlated in the absence of outliers in the time series. In order to avoid generating a group of highly correlated features, a principal component analysis (PCA) [15] is carried out and this step will help to select relevant features that are de-correlated.

Results and Discussion

In order to demonstrate the effectiveness of FAST, a case study is presented in this section. Given the set of time series obtained from [7], two sets of features are obtained:

1) one set with just the raw data and
2) one set using FAST.

These two sets of features are used separately to facilitate the training of two individual 2-layer neural network (NN) models [15]. A holdout cross-validation [16] is used for training and testing. Approximately 60 % and 40 % of data is used as the training set and testing set, respectively. Similar to that of described in [7], the models attempt to classify abnormal cooking conditions based on the available sensor data.

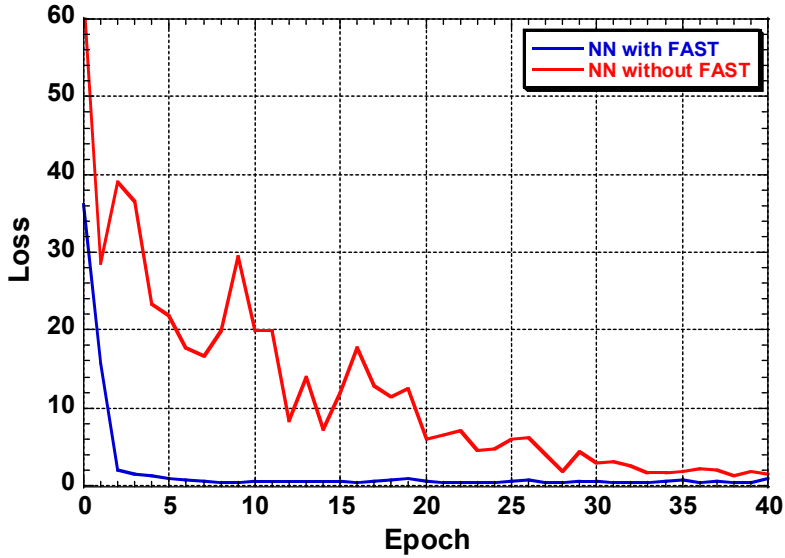

Fig. 4. Loss for the two NN models during training.

For comparison, the two NN models are trained to have the same level of prediction accuracy[4]. Fig. 4 show the loss[5] as a function of epoch for two NN models. It can be seen that the model using FAST converges within 10 epochs and the model using raw data as features requires about four times more epochs to reach convergency. This highlights the benefits of obtaining relevant features for efficiency in training.

Also, although both models are with 2-layer NN architecture, the NN model with FAST needs only 13 neurons for the first layer and 6 neurons for the second layer whereas the NN model without FAST requires 240 neurons in the first layer and 120 neurons in the second layer in order to achieve same level of prediction accuracy. It is worth noting that the prediction capability can generally be enhanced with the increasing number of layers and neuron. However, the model is highly possible to suffer from overfitting [17] in which the model is learning trends and patterns that are not used (i.e. the data oscillation as seen in Figs.1). Therefore, FAST can help model developers to avoid such training problems.

Conclusion and outlook

In this paper, the automated feature engineering tool, FAST (Feature extrAction and Selection for Time-series), is introduced. The overall structure of FAST and its operation procedure are described. Given a dataset similar to that of found in [7], 10 most-relevant features are obtained using FAST. The suggested features are then used in a case study to demonstrate its effectiveness. Results show that a machine learning model, such as a 2-layer neural network, trained using features obtained from FAST is more numerically efficient in terms of model training and has much simpler model structure. These observations highlight the importance of feature engineering. It is shown that the use of FAST will facilitate the development of an efficient and accurate machine learning model in fire research community.

Acknowledgement

The authors would like to thank Dr. Michael Huang and Dr. Wei Tang for the helpful discussion.

References

[1] Cao, L. J., & Tay, F. E. H. (2003). Support Vector machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Transactions on Neural Networks, 14(6), 1506-1518.

[4] The accuracy is defined as the corrected predicted label over all possible labels.
[5] Loss value implies how well or poorly a certain model behaves after each iteration of optimization.

[2] Anton, S. D., Ahrens, L., Fraunholz, D., & Schotten, H. D. (2018, November). Time is of the Essence: Machine Learning-based Intrusion Detection in Industrial Time Series Data. In 2018 IEEE International Conference on Data Mining Workshops.

[3] Kevin McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., & Vanella, M. (2018). Fire Dynamics Simulator User's Guide. NIST Special Publication, 1019, 1.

[4] Mensch, A., Hamins, A., Tam, W.C., Lu, J., Markell, K., You, C., & Kupferschmid, M. Sensors and Machine Learning Models to Prevent Cooktop Ignition and Ignore Normal Cooking. Submitted to Fire Technology.

[5] Asuncion, A., & Newman, D. (2007). UCI Machine Learning Repository.

[6] Mensch, A. E., Hamins, A. P., Lu, J., & Tam, W. C. (2019). Evaluating Sensor Algorithms to Prevent Kitchen Cooktop Ignition and Ignore Normal Cooking. SUPDET, Denvor, CO.

[7] Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests. *Neurocomputing*, *307*, 72-77.

[8] Tavenard, R. (2017). tslearn: A machine learning toolkit dedicated to time-series data. URL https://github. com/rtavenar/tslearn.

[9] Brown, C. U., Vogl, G. W., & Tam, W. C. (2019). Measuring Water Flow Rate for a Fire Hose Using a Wireless Sensor Network for Smart Fire Fighting. SUPDET, Denvor, CO.

[10] Stojanova, D., Panov, P., Kobler, A., Džeroski, S., & Taškova, K. (2006). Learning to Predict Forest Fires with Different Data Mining Techniques. In Conference on Data Mining and Data Warehouses, Ljubljana, Slovenia.

[11] Rice, J. A. (2006). Mathematical statistics and data analysis. Cengage Learning.

[12] Massey Jr, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. Journal of the American statistical Association, 46, 68-78.

[13] Kendall, M. G. (1938). A new measure of rank correlation. Biometrika, 30(1/2), 81-93.

[14] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.

[15] Ruck, D. W., Rogers, S. K., & Kabrisky, M. (1990). Feature selection using a multilayer perceptron. Journal of Neural Network Computing, 2(2), 40-48.

[16] Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics surveys, 4, 40-79.

[17] Burnham, K. P., & Anderson, D. R. (2002). Model selection and multimodel inference. Springer Verlag. New York.