NETWORK TRAFFIC ANOMALY DETECTION USING EMD WITH ANN AND LSTM

Muzaffer Mert Guneri*, Wenqin Shao†, Frank Brockners†
*Telecom ParisTech
muzaffer.guneri@telecom-paristech.com
†Cisco systems
{wenshao, fbrockne}@cisco.com

***Abstract*—Anomaly detection in network traffic is an important issue in which there are many methods. However, it is not easy to find a precise boundary between normal and abnormal behaviour. This paper focuses on whether the detection can be improved solely via proper data preprocessing method so that noises are suppressed and useful information is extracted. Since network traffic is generally sampled at a certain time interval, we can use time-series methods which help us to find pattern in our dataset. EMD (Empirical Mode Decomposition) is a powerful tool for network traffic signals as it handles non-stationary and non-linear data. EMD can also work online which is important to detect anomalies timely. In this paper, the use of EMD in the preprocessing of general detectors(Artificial Neural Network,Long short-term memory) is compared with the use of the original dataset.**

***Index Terms*—Anomaly Detection, Empirical Mode Decomposition (EMD), Artificial Neural Network (ANN), Long short-term memory (LSTM), Intrinsic Mode Function (IMF)**

I. Introduction

With the tremendous growth of computer network usage, it has become important to detect anomalies in the network. Anomaly detection is a technique used to identify rare data points and events that do not match the pattern. Anomalies usually occur infrequently and might be related to major and significant threats, such as attacks, hardware issues or fraud actions. It has become a widely studied subject to detect anomalies timely and accurately. However, it is not easy to distinguish anomalies from normal network actions because data might contain noise which might be similar to anomaly event in case boundary between normal and anomaly behavior is not precise.

Many solutions have emerged in the past years on this issue. Although there are a lot of technical and scientific studies on anomaly detection methods for network traffic, studies like density based algorithm and clustering algorithms [1] cannot use the information in the sequence. In addition, some of the solutions focus on stationary data such as solutions with Fourier Transformation [2]. Since network traffic is generally sampled at a certain time interval, it can be used as time-series data with algorithms which have knowledge of sequence. In addition, the stationarity of network traffic is just an assumption, we cannot make sure that network traffic is stationary [3]. A signal is called stationary if it's statistics don't change over time. Otherwise, it is non stationary.

Our main motivation is to find something in our preprocessing part to extract relevant and important information which is useful for anomaly detection from our original time series data, then to check whether we can improve detection accuracy. In order to do that we have to look at methods which can work online, with time-series and with non-stationary data.

EMD(Empirical Mode Decomposition) is a decomposition method which can be used online [4]. Apart from being online, we have two main reasons in order to choose EMD in our preprocessing part.

First reason is that EMD gives us information about frequency. Anomalies are unexpected or extreme, more importantly rare values and these things affect frequency domain [5]. Decomposition methods can extract these rare values and can show these changes because decomposition methods like EMD and Fourier give us knowledge of frequency. EMD decomposes a signal into so-called Intrinsic Mode Functions (IMF) along with a trend. IMFs represent the frequency and amplitude characteristics of a signal [6].

Second reason is that EMD can be used with non-stationary and non-linear signals. Since the decomposition is based on the local characteristics of the data, it can be used on a non-stationary signal. Fourier Transform is one of the most popular methods, but the best decomposition technique to apply is not clear when a signal is non-stationary, as would be in our case. EMD is one alternative to Fourier decomposition in which the components of a signal are not fixed in frequency over time.

Artificial Neural Network (ANN) and Long short-term memory (LSTM) are used as generic detectors. Both ANN and LSTM are robust dynamic classifiers. In addition, LSTM is used because it has knowledge of sequence which is important for time-series data. With this information, naturally LSTM is expected to work coherently with EMD. ANN is used to determine whether we can improve results even though algorithm does not have knowledge about sequence only dealing with data point by point.

In this paper, signal processing technique, EMD, and machine learning techniques, ANN and LSTM, are jointly used. EMD is used for the transformation of input time-series data in preprocessing part of both LSTM and ANN. Detection accuracy of using original time-series features (without EMD) is compared with using EMD in section VI. Our contribution is to prove that EMD works well with time-series data for

anomaly detection and that EMD can improve algorithms' detection accuracy. The accuracy increased by almost 9%. The paper is organized as follows: section II is related works. Section III is introduction of datasets. Background of EMD method, ANN, LSTM are described in section IV. How the model works is described in section V. Section VI shows the results. Section VII is the conclusion part

II. Related Work

Many solutions have emerged in the past years on anomaly detection such as signal processing techniques and machine learning techniques. Examples of signal processing techniques include wavelet analysis [7], EMD and Hilbert-Huang transform [8]. Examples of machine learning techniques include LSTM [9].

Jieying Han introduced three methods for internet traffic anomaly detection based on three signal characteristics [8]. First one is Hurst parameter which is calculated based on the first Intrinsic Mode Function (IMF). Hurst parameter is expanded by introducing a weighted self-similarity based on the concept of entropy. They used KDD99 dataset which is also used in this paper. They used one time-series feature in their experiments. The main problem is that Hurst parameter cannot be used online because it is a measurement of long-term memory of time series.

Xiaorong C. et al. proposed Anomaly detection based on self-similarity using EMD and Wavelet transform [7]. Their study looks like ours. They tried to improve result with using EMD. However, they chose Hurts parameter in order to detect anomalies and Hurst parameter has issues in online.

Staudemeyer's research focused on LSTM [9]. They investigated suitable parameter and structure for LSTM while working with KDD99 dataset. Their aim was to find the most efficient LSTM structure. In our cases we are also using LSTM but our aim is not finding the best LSTM structure, we want to improve LSTM result for all general structure. Although their work is on LSTM, some of the features that they used are not time-series.

III. Dataset

A. KDD99

It is one of the most popular datasets which is generally used for anomaly detection system. KDD dataset consists of single connection vectors each of which contains 41 features and is labeled as either normal or an attack [10]. A connection is a sequence of TCP packets starting and ending at some well defined times. There are four different attack categories. It has traffic features computed using a two second time window. We have focused on these features(time-series features) in KDD99. Dataset is avaible at [11]. It is important to note that the test data is not from the same probability distribution as the training data, and it includes specic attack types not in the training data which make the task more realistic

We used KDD10% as our train data and we used full test data as our test data. We labeled all attack types as one event so we have two labels(attack and normal) at the end. You can see how the values change over time during an attack in figure 1. Red dots show the attack data point and also figure 1 shows that this feature is a time series data.

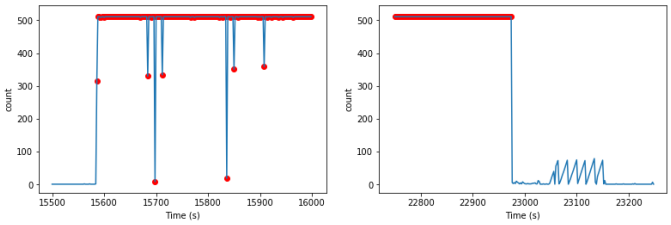

Fig. 1. Beginning and Ending of Smurf attack(Count: number of connections to the same host as the current connection in the past two seconds)

Table 1 shows the event distribution and also the number of status change (attack to normal or normal to attack event). We can easily see that we have a proportional number of status change in the train, but in the test dataset we have too many status changes. The reason is that they wanted to see how they would be able to give some difficulty to the algorithms and test them even under these conditions.

TABLE I
Dataset Distribution

Dataset	Normal Datapoints	Attack Datapoints	Status Change
10%KDD	97277	396743	533
Test dataset	60592	250436	19178

IV. Background of Generic Detectors and EMD

A. EMD(Empirical Mode Decomposition)

It is a decomposition method of breaking down a signal without leaving the time domain. EMD decomposes a signal into a sum of IMFs. An IMF is a function that:

- Has only one extrema between two succesive zero crossings
- At any point, the mean value of the upper envelope and the lower envelope is zero.

The first condition is similar to the traditional narrow band requirements for a stationary Gaussian process. The second condition is new, its locality is necessary so that the instantaneous frequencies will not have unwanted fluctuations induced by asymmetric waveforms [12].

EMD uses Sifting process in order to extract IMFs from signal. Figure 2 shows the how the Sifting Process works. The sifting process is as follows:

Given a signal X(t):

1) Identify all local extrema values
2) Generate upper and lower envelopes using cubic spline interpolation [13] with local extrema values.
3) Calculate the mean of lower and upper envelopes(m = point by point mean from upper and lower envelopes).
4) Substract the mean from the signal.$h(t) = X(t) - m(t)$

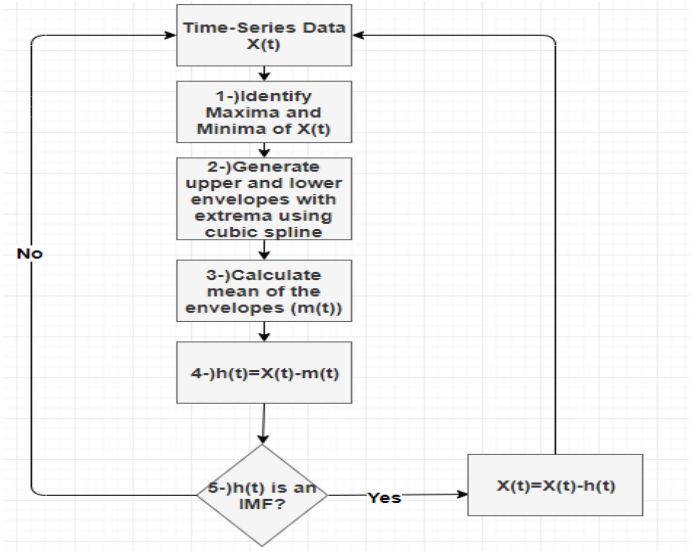

Fig. 2. Sifting Process

5) If h satisfies IMF conditions than replace X(t) with residual $r(t) = X(t) - h(t)$. If h does not satisfy conditions, replace X(t) with h(t).
6) Repeat previous steps until residual is met stopping criteria.

There are more than one stopping criteria:

1) The constant criterion sets a constant iteration number for the sifting process.
2) Stops the sifting process when the number of zero crossing and extrema stays constant for S successive iterations
3) The stopping criterion can be the Standard Deviation (SD) between two consecutive results in the sifting process. If the SD value is smaller than the pre-defined parameter, the iteration will be stopped. In most studies, the pre-defined parameter was used as 0.3.

At the end of the process, signal can be expressed as where $c_i(t)$ is the i^th IMF and r_n is the residual:

$$X(t) = \sum_{i=1}^{n} c_i + r_n \tag{1}$$

B. ANN(Artificial Neural Network)

Neural networks consist of input layer, output layer and hidden layer consisting of units that transform the input into something that the output layer can use. Artificial neural network is not a new concept. In 1943, it was named as perceptron which is made of McCulloch-Pitts neurons. However, it is only popular in the last several decades. This is due to the arrival of a technique called backpropagation which allows networks to adjust their hidden layers of neurons in situations where the outcome doesnt match what the creator is hoping for.

Figure 3 shows a multilayer perceptron. It has one input layer, one hidden layer and one output layer. w_i represents

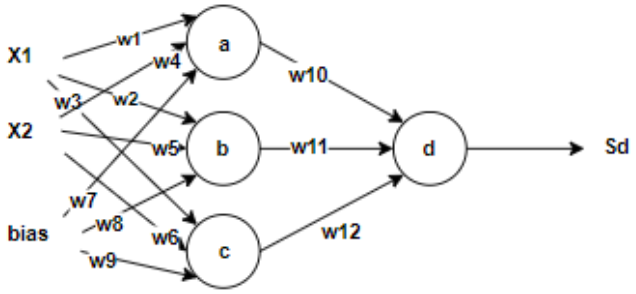

Fig. 3. ANN-Multilayer Perceptron

weights and S represents results of activation function. Backpropagation can be summarized as follows:

Let say we have a two signals x1, x2 and bias = 1, $1 \leq i \leq n$ and n is the lenght of the signals. Firstly input values multiply with weights then sum for each layer. For node a we can show as such:

$t_a = X_{1i}.w_1 + X_{2i}.w_4 + w_7$

Then activation function such as sigmoid is used.

$s_a = sig(t_a) = \frac{1}{1+e^{-t_a}}$

Same things are done for the next layer.

$t_d = s_a.w_{10} + s_b.w_{11} + s_c.w_{12}$

$s_d = sig(t_d) = \frac{1}{1+e^{-t_d}}$

s_d is our output. In backpropagation algorithm, we choose a loss function in order to calculate our error such as mean squared error.

$error = E = 1/2(target - s_d)^2$

for the backpass in backpropagation

$\frac{\partial E}{\partial w_{10}} = \frac{\partial E}{\partial s_d}.\frac{\partial s_d}{\partial t_d}.\frac{\partial t_d}{\partial w_{10}} = (-(target - s_d)).(s_d.(1 - s_d)).s_a$

New w_{10} value is $w_{10}^{new} = w_{10}^{old} - \frac{\partial E}{\partial w_{10}}$

C. LSTM (Long short-term memory)

A LSTM network is a kind of recurrent neural network. A recurrent neural network is a neural network that attempts to model time or sequence dependent behavior. This is performed by feeding back the output of a neural network layer at time t to the input of the same network layer at time t + 1. Recurrent neural network has problems like vanishing gradient problem and long-term dependencies. Detail analysis about these problems is presented in [14]. LSTM can solve this problem because it uses gates to control the memorizing process [15].

Figure 4 shows a LSTM cell. Pink circles are pointwise operations and rectangles are neural network layers. σ : Sigmoid layer, sigmoid layer output can be 0 or 1 which is perfectly good for forgetting or remembering. X(t) is current input, h(t-1) is output of last LSTM unit, c(t-1) is memory from last LSTM unit.

The sigmoid layer takes the input X(t) and h(t-1) and helps the network learn which state variables should be remembered or forgotten. This gate is called forget gate f(t). b is the bias and W is the weight of related layer.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2}$$

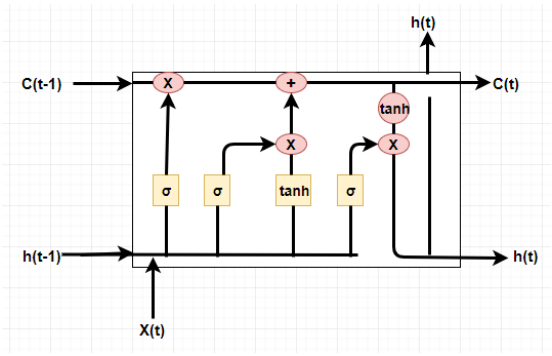

Fig. 4. LSTM Cell

The next step is to decide what new information we are going to store in the cell state. The first step for this combined input is for it to be squashed via a tanh layer. The second step is that this input is passed through a Sigmoid layer(this step is also called input gate i(t)). An input gate is a layer of sigmoid activated nodes whose output is multiplied by the squashed input. This new memory is then added to old memory c(t-1) to give c(t).

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{3}$$

$$C_t = [f_t * C_{t-1} + i_t] * [tanh(W_C.[h_{t-1}, x_t] + b_C)] \tag{4}$$

In final step, which parts of the cell we are going to use as the output(which is also called output gate o(t)) is decided by the sigmoid layer. Then, we put the cell state through a tanh generating all the possible values and multiply it by the output of the sigmoid gate in order to use the parts that we decided on.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

V. Proposed Approach

This section focuses on whether EMD can improve the results. All experiments have been done with KDD99. Four time-series features were chosen (count, srv-count, dst-host-count, dst-host-srv-count) in datasets. These are traffic features computed using two second time window. More information can be found here [11].Figure 5 shows our approach.

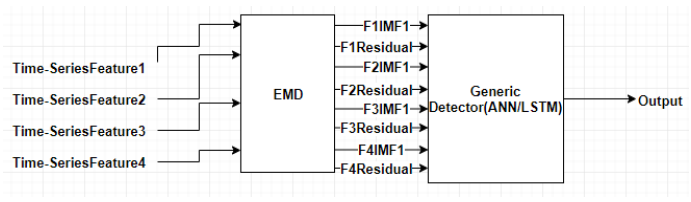

Fig. 5. Model

First, four time-series features are decomposed into first IMF using EMD method. For each feature, we extract the first IMF and the residual. There are two main reasons for using the first IMF. First reason is that EMD extracts IMFs with decreasing order which means the first IMF has the highest frequency in the signal. We expect to see the most repeated pattern in the first IMF. Secondly, We want to do a comparison between using EMD and without EMD, so we want to see same affect for each feature but if we dont specify a constant IMF number, feature might give us different number of IMF and their affect on the output would be different. If all signal is attempted to decompose, it takes too much time and it is not a usable case for anomaly detection because it is expected in the online version. There is a solution for the online version of EMD [4]. With that online solution, signal is divided into windows. Since IMF values depend on local extrema, we can use windows and get the same values. However the only difference can occur in the border and that depends on the window size. Window size is used as 100 samples in this experiment. Some of the windows could not extract more than 1 IMF. If some features extract 1 IMF and some of them extract more than 1 IMF than their effects will not be the same for the result. Since we want to compare results, we want to see equal effects from all features.

There are two methods in generic detector. First method is ANN. If EMD is used, then input nodes are eight because we have the first IMF and the residual for all features. For without EMD or only first IMF, we have four time-series features. ANN model has two hidden layers and each of them has 30 nodes. Each hidden layer is connected to dropout in order to reduce overfitting. Leaky Relu is used as activation function between hidden layers. We have two labels: 1 for normal event, 0 for attack event. So in output layer, sigmoid function is used as activation function because Sigmoid function is the most popular function for binary class classification(in our example attack or normal).

$$LeakyRelu(x) = max(x * 0.01, x) \tag{7}$$

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

Label distribution is not equal in KDD99 dataset, so mean squared error is not a good option for loss function. Cross entropy is one alternative of mean squared error as a loss function in case target distribution is not equally distributed. In our cases there are two events, so binary cross entropy is used as loss function. ANN model with EMD is shown in figure 6.

$$BinaryCrossEntropy = -(ylog(p) + (1-y)log(1-p)) \tag{9}$$

Second generic detector is LSTM. If We choose to use first IMF and residual then input nodes are eight. With only first IMF or without EMD there are four time-series features. We have one LSTM layer and size of the hidden state of an LSTM unit is 32. We also have dropout in order to prevent over-fitting. Sequence size for each row is the same as window size which is 100. Binary Cross entropy is used as loss function and sigmoid function is used as activation function in the output. LSTM model with EMD is shown in figure 7. Each input is a vector with lenght of 8.

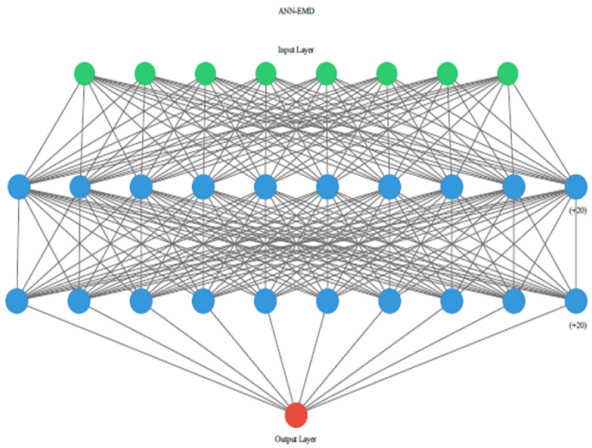

Fig. 6. ANN model With EMD

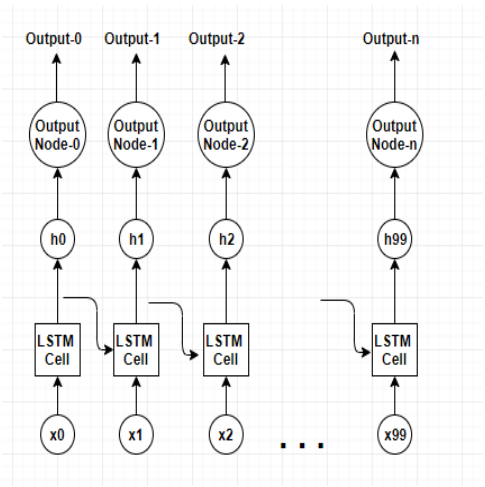

Fig. 7. LSTM model

VI. Experiment And Analysis

A. Performance Measure

The following conclusions are formulations which are derived from the confusion matrix (Table II). They are explained through a small example below.

$Accuracy = \frac{TP+TN}{TP+TN+FN+FP}$

Recall: It shows how well the positive cases are estimated.

$Recall = \frac{TP}{TP+FN}$

Precision: A situation that shows success in a positively predicted situation.

$Precision = \frac{TP}{TP+FP}$

F-Measure(F-score): Harmonic average of Recall and Precision.

$FMeasure = \frac{2*Precision*Recall}{Precison+Recall}$

B. Result and Analysis

1) Pointwise Evaluation: Normally each data point(connection) has a label. We considered each data point with pointwise evaluation and we tried to find out whether the label is normal or attack for each data point in the test.

TABLE II
Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	**TN(True Negative)**	**FP(False Positive)**
Actual Positive	**FN(False Negative)**	**TP(True Positive)**

This paper's aim is to show the affects of EMD-IMF. General expectation is the improvement in results especially with LSTM because it has knowledge of sequence. We also used ANN in order to show impact of EMD on the results of algorithms that deal with data point by point. Table III shows the experiment results of using EMD with ANN and LSTM. It shows the results of using first IMF as train data and also using first IMF and residual of original signal together as train data.

Results are quite satisfying for all cases (Accuracy,precision,f-measure) except for recall because it is already pretty high (around 98%). With original data(without using EMD), precision is around 50%, so actually half of the normal predicted values are anomalies. However, if we use EMD, precision increases around 10%. Label distribution of KDD 10 percentage dataset is not equal. Almost 80% of the data is attack so we should not look and compare our efficiency of algorithm with accuracy but we can look at F-measure and we can see at least almost 5% improvement. It is pretty obvious that if we use EMD, results are improving for both algorithms(ANN-LSTM) and different models.

We used different types of models (different node number,hidden layer number,activation function) for both algorithms (ANN/LSTM). When we compare results between First IMF and First IMF with Residual, we saw two different outcomes. For ANN, IMF with Residual solution gives better results. On the contrary, First IMF solution gives better results for LSTM. One of the reasons why LSTM works better with only first IMF might be that the first IMF provides sufficient information to detect the attacks and LSTM has already the sequence information. It looks like an extra thing when we give two series which extracted from same time-series. ANN works better with Residual because it does not have knowledge of sequence and residual gives aspect of all data. However, in LSTM aspect, First IMF already has enough information about critical changes and residual has extra information that does not needed for LSTM.

EMD improved every side (except for recall which is already good enough). Recall values show us that the generic detectors can detect normal behavior easily with or without EMD. However, they have issues with detecting abnormal behavior. Precision values show us that EMD helps generic detector to detect abnormal behavior more precisely. Due to LSTMs definition, it is expected to work very well with EMD and results show that EMD extracts IMFs and these IMFs have

TABLE III
RESULTS

	ANN	EMD-ANN (First IMF)	EMD-ANN (IMF and Residue)	LSTM	EMD-LSTM (First IMF)	EMD-LSTM (IMF and Residue)
Accuracy	**0.8087**	**0.8789**	**0.8985**	**0.8405**	**0.8841**	**0.8727**
Recall	**0.9836**	**0.9803**	**0.9808**	**0.9711**	**0.9725**	**0.9727**
Precision	**0.5047**	**0.6193**	**0.6612**	**0.5514**	**0.6314**	**0.6081**
F-measure	**0.6670**	**0.7591**	**0.7899**	**0.7034**	**0.7657**	**0.7483**

sensitive and important information for finding anomalies in the signal.

2) Eventwise Evaluation: We tried to look at the successive data points which have the same label in that evaluation. We have separated all the events with status changes. Therefore, we tried to find out only the status changes in eventwise evaluation but we had some serious issues. LSTM algorithm's prediction is showed in figure 8. one is a normal event and zero is an attack event in figure 8. Blue colour shows the test data and yellow colour shows the prediction. We realized that test dataset is time-series but with very different temporal characteristics. Figure 8 shows that in some cases there is an attack on only one data point, then a normal data point and it continues in a similar way. This interleaved normal and abnormal data points bring a very different sequence pattern. LSTM algorithm seemingly detects the event very well if we look at it intuitively. However, there a lot of data points that LSTM could not detect well. That is why when we try to detect status changes we got pretty low precision and recall.

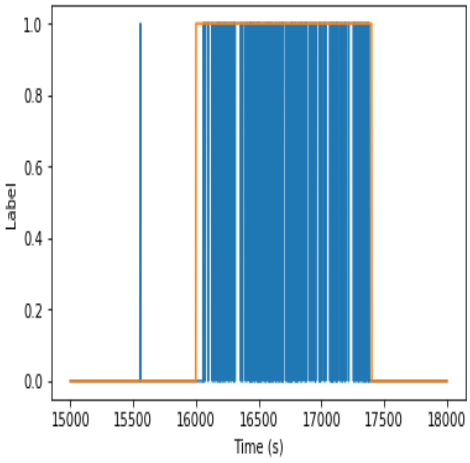

Fig. 8. Eventwise Model Representation

VII. CONCLUSION

We proposed a model which uses IMF and Residual instead of using original data/signal for different algorithms. This paper's contribution is to proved that EMD works well with time-series data for anomaly detection and EMD can improve algorithms detection rate. The experimental results show that boundary between normal and anomaly behavior is more precise with IMF values rather than original data. Therefore we can detect anomalies more accurately with EMD. For future work; since IMFs are extracted, Hilbert transform can be applied for these IMFs. After Hilbert transform, the instantaneous frequency, marginal spectrum and energy density level can be used instead of IMFs.

REFERENCES

[1] Singh, Satinder & Kaur, Guljeet. (2007). Unsupervised Anomaly Detection In Network Intrusion Detection Using Clusters.

[2] Jiang, Dingde & Xu, Zhengzheng & Zhang, Peng & Zhu, Ting. (2013). A transform domain-based anomaly detection approach to network-wide traffic. Journal of Network and Computer Applications. 40. 10.1016/j.jnca.2013.09.014.

[3] Z Wang and W Li, Research on the network traffic time series modeling and forecasting based on wavelet decomposition, Journal of Convergence Information Technology 7 (2012), 124131.

[4] R. Fontugne, P. Borgnat and P. Flandrin, "Online Empirical Mode Decomposition," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 4306-4310. doi: 10.1109/ICASSP.2017.7952969

[5] Zimek, Arthur; Schubert, Erich (2017), "Outlier Detection", Encyclopedia of Database Systems, Springer New York, pp. 15, doi:10.1007/978-1-4899-7993-3_80719-1, ISBN 9781489979933

[6] N.E. Huang, Z. Shen, S.R. Long, M.e. Wu, H.H. Shih, Q. Zheng, N.e. Yen, e.e. Tung, and H.H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," Proceedings of the Royal Society of London. Series A, vol. 454, no. 1971, pp. 903-995, 1998.

[7] X. Cheng, K. Xie, D. Wang , "Network Traffic Anomaly Detection Based on Self-Similarity Using HHT and Wavelet Transform ", 2009 Fifth International Conference on Information Assurance and Security, August 2009

[8] Jieying Han, "NETWORK TRAFFIC ANOMALY DETECTION USING EMD AND HILBERT-HUANG TRANSFORM"(master's thesis),Western Carolina University, USA, 2013.

[9] : R. C. Staudemeyer and C. W. Omlin, Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data, in Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference. ACM, 2013, pp. 218-224.

[10] :M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, in Computational Intelligence for Security and Defense Applications, Ottawa, pp. 1-6, 2009

[11] :KDD99 data set for network-based intrusion detection systems. December 2018. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[12] : J. Z. Zhang, B. T. Price, R. D. Adams and T. J. Knaga, Detection of Involuntary Human Hand Motions Using Empirical Mode Decomposition and Hilbert-Huang Transform, in Circuits and systems, MWSCAS 2008. 51st Midwest Symposium, pp. 157160, Aug. 2008.

[13] :Cubic Spline Interpolation. January 2019. [Online]. Available: https://mse.redwoods.edu/darnold/math45/laproj/Fall98/SkyMeg/Proj.PDF

[14] : S. Hochreiter, Y. Bengio, P. Frasconi and J. Schmidhuber. Gradient ow in recurrent nets: The diculty of learning long-term dependencies. In A eld guide to dynamical recurrent neural networks. IEEE Press, 2001.

[15] :S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, vol. 9, no. 8, pp. 1735 1780, 1997. DOI http://dx.doi.org/10.1162/neco. 1997.9.8.1735.