NADINE's Supplemental Document

ACM Reference Format:
. 2019. NADINE's Supplemental Document. In *Proceedings of CIKM*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

This document provides additional materials of our paper titled "Self Construction of Multi-layer Perceptron Networks in Lifelong Environments". It contains dataset description and pseudo-code of NADINE's learning modules.

1 DATASET DESCRIPTION

This section outlines characteristics of nine datasets made use of our paper titled "Self Construction of Multi-layer Perceptron Networks in Lifelong Environments". The nine datasets are detailed as follows:

- *SUSY Problem*: SUSY dataset [1] is a very popular dataset in the area of big data with five million samples. It is commonly used for classification task with two classes problem. The first eight features represent the kinematic properties of signal process whereas the remaining features are generated from the function of the first eight features. This dataset is not categorized as a nonstationary dataset. However, due to the big size of the dataset, this dataset enables the algorithm to demonstrate the ability to handle the lifelong learning environments.
- *Hepmass Problem*: Like Susy, Hepmass dataset is also a prominent in the big data area. It features a two-classes classification task which aims to separate particle-producing collisions from a background source. The dataset consists of 28 input attributes, where the first 22nd features are low-level features. The next 5th features are high-level features and the last feature is a mass feature. There are 10500000 samples in total contained in the dataset. In our experiment, we utilize only 2 million samples (around 19 percent of the total data) with 2000 time stamps.
- *Rotated MNIST Problem*: The rotated MNIST [6] is a popular continual learning problem developed from the original MNIST problem [5]. It applies rotation of original MNIST problem with randomly generated angels in between $-\pi$ to π inducing abrupt drift. To realize challenging simulation environment, overall data samples are grouped into 65 tasks containing 1000 samples and simulated under the prequential test-then-train protocol examining the generalization performance without feeding a model with relevant samples. Unlike benchmark setting, this procedure allows to obscure the task's boundary and time points of concept changes.
- *Permuted MNIST Problem*: The permutted MNIST [7] is another benchmark problem derived from the original MNIST dataset [5]. Random permuttation is implemented in the input pixels resulting uncorrelated input distributions across different data concepts. As with the rotated MNIST problem, it is formed as 70 tasks containing 1000 samples. Three random permutations are realized and lead to three different concepts with unknown task boundaries - each task may be drawn from the same or different concepts. Our simulation is carried out by following the prequential test-then-train simulation protocol.
- *KDDCup Problem*: KDDCup dataset [8] introduces the intrusion detection problem in the form of two classes classification tasks. The task aims to recognize whether the network connection is under the attack condition or not. This problem presents the non-stationary components due to the existence of the various type of intrusions held in military network environment. This dataset is very popular dataset as it was used for Third International Knowledge Discovery and Data Mining Tools Competition. In this experiment, we use only 10% of the total data (500K) for our numerical study with 100 time stamps under the test-then-train procedure.
- *SEA Problem*: The SEA is an artificial dataset [3] which features two classes classification problem. It contains three input features, where the first two features are relevant features and the third feature acts as a noise. A sample is classified as a class 1 if the following condition $f_1 + f_2 < \theta$ is satisfied. Otherwise, the sample is classified as a class 2. The transition of the class threshold three times $\theta = 4 \longrightarrow 7 \longrightarrow 4 \longrightarrow 7$ triggers the two types of drift : abrupt and recurring. The original SEA dataset features have values between zero to ten. However, in this case, we utilize the modified version of SEA problem used by [4] which has a class imbalanced problem properties with 5% to 25% class proportion. The total sample of this dataset is 200K. The SEA dataset is very crucial in developing a controlled simulation environment where the type of drift and the time instant when the concept drift occurs are known. These two classes output is determined based on d-dimensional random hyperplane $\sum_{j=1}^{d} w_j x_j > w_o$. For this dataset, we conduct the experiment under prequential test-then-train process with 100 time stamps.
- *Household Electricity Consumption Problem*: The household electricity consumption problem describes a regression problem of electricity load of an individual household located in Sceaux, France recorded from the period of December 2006 and November 2010 with a sampling rate of one minute. It consists of 2075259 instances and seven input attributes. The electricity load is measured in three different sections of a house creating the multi-input-multi-output (MIMO) problem. This problem contains seasonal changes following peak and off-peak patterns of electricity usage.

CIKM, November 3rd-7th 2019, Beijing, China

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Algorithm 1 Learning policy of NADINE

Input: Data Stream : $[B_1, B_2, ..., B_k, B_K]$, $I \longleftarrow$ dimension of input B_k
Execute the main loop process:
for $k = 1$ to K (the number of data batches) **do**
  Obtain: input k-h data batch $X_k \in \Re^{T\times n}$
  Testing phase:
  Execute:
  - feedforward operation
  - Output of k-h data batch $O_k \in \Re^{T\times m}$
  Training phase:
  Obtain:
  - Target data label $C_k \in \Re^{T\times m}$.
  - Obtain accuracy matrix A (Compare Y_k Vs C_k); $A \in \Re^{T\times 1}$
  Perform:
  (1) Calculation of the correlation between the hidden layers output and the softmax output. **Algorithm (3)**.
  (2) Drift Detection Method Based on the Hoeffding's inequality. **Algorithm (2)**
  for $t = 1$ to T (the size of a data batch) **do**
    Execute: feedforward operation
    Calculate: $e_t = X_t - z_t$, $\mu^t_{\tilde{X}}$, $\sigma^t_{\tilde{X}}$, $E[z]$, and $E[z^2]$
    Calculate: μ^t_{Bias}, σ^t_{Bias}, μ^t_{Var}, and σ^t_{Var} utilizing $E[z]$ and $E[z^2]$
    Hidden node growing mechanism:
    if $(\mu^t_{Bias} + \sigma^t_{Bias} \geq \mu^{min}_{Bias} + \pi\sigma^{min}_{Bias})$ **then**
      - $K = K + 1$ (K is the number of node in the last hidden layer)
      - *Initialization*: random weight parameter of added node.
      - *set*: $grow = 1$
    else
      - *set*: $grow = 0$
    end if
    Hidden node pruning mechanism:
    if $(\mu^t_{Var} + \sigma^t_{Var} \geq \mu^{min}_{Var} + 2\chi\sigma^{min}_{Var})$ && $(grow = 0)$&&$(t > I + 1)$&&$(K > 1)$ **then**
      for $i = 1$ to R **do**
        Calculate: HS
      end for
      Prune: hidden node which has the smallest value of HS
      $K = K - 1$
    end if
    Execute: backpropagation
    Update network parameter
  end for
  Execute:
  (3) Adaptive memory storage mechanism. **Algorithm (4)**
end for

Algorithm 2 Drift Detection Method Based on Hoeffding Bound Concept

Input: Accuracy matrix $A \in \Re^{T\times n}$, $B \in \Re^{cut}$,ϵ_{drift},$\epsilon_{warning}$
Output:$C \in \Re^{(T-cut)}$, Status : Drift, Warning, Stable
for $k = 2$ to K (the number of data batches) **do**
  Determine Switching Point (Cutting Point)
  Obtain : $\tilde{A}$ and $\tilde{B}$:
  for $t = 1$ to T (the size of a data batch) **do**
    Calculate Hoeffding Error Bound $\epsilon_{\tilde{A}}$, $\epsilon_{\tilde{B}}$ formula (8)
    if $(\tilde{A} + \epsilon_{\tilde{A}} \leq \tilde{B} + \epsilon_{\tilde{B}})$ **then**
      $cut = t$
      Obtain: $\tilde{C} \in \Re^{T-cut}$; break
    end if
  end for

  if $|\tilde{B} - \tilde{C}| > \epsilon_{drift}$ **then**
    (1) Add New Layer; (2) Feed: input data+class label ($[B_k Y_k]$) + data in adaptive memory + data in the buffer ($[B_{k-1} Y_{k-1}]$); (3) Then remove data in the adaptive memory and the buffer after feeding the data
  else if $\epsilon_{warning} \leq |\tilde{B} - \tilde{C}| < \epsilon_{drift}$ **then**
    (1) Create data in the buffer ($[B_k Y_k]$); (2) Feed: input data+class label ($[B_k Y_k]$)
  else
    (1) Feed: input data+class label ($[B_k Y_k]$); (2) Remove data in the buffer
  end if
end for

Algorithm 3 Calculation of the correlation between the hidden layers output and the softmax output

Input: Network Structure, $D \longleftarrow$ number of hidden layer, $R_d \longleftarrow$ number of hidden nodes of d-th hidden layer, $m \longleftarrow$ number of class output
Output: Scorr (Correlation Rate), Learning rate (The importance of hidden layer to the output)
for $k = 1$ to K (the number of data batches) **do**
  if $D > 1$ **then**
    for $i = 1$ to D **do**
      for $j = 1$ to R **do**
        for $o = 1$ to m **do**
          Obtain: Correlation R_j and m_o
        end for
      end for
      Obtain: Matrix Correlation:**HnCorrOut(i)**$\in \Re^{j\times o}$
      Obtain: $Scorr(i)$ obtained from mean of mean **HnCorrOut(i)**: $\overline{\overline{HnCorrOut(i)}}$
    end for
    Obtain **learningRate** $\longleftarrow 0.01 * exp(-(1./Scorr - 1))$
  end if
end for

- *Condition Monitoring Problem*: this problem [2] presents regression problem of naval propulsion plant condition generated from Gas Turbine (GT) simulator of a frigate. It comprises 11934 data points and 16 input attributes. There exists two target variables GT Compressor decay state coefficient and GT Turbine decay state coefficient. This problem describes the decaying effect of components as the unique trait of the condition monitoring problem.

Algorithm 4 Adaptive Memory Storage Mechanism

Input: Data stream chunk $B_k \in \Re^{T \times n}$,$ca \leftarrow n+1$, $\alpha_1 \leftarrow 0.99$, $\alpha_2 \leftarrow 0.999$, $\delta \leftarrow 0.55$
Output: Selected samples from B_k
for $t = ca$ to T (the size of a data batch) **do**
  if $t \geq ca$ **then**
    Obtain mean and inverse covariance matrix recursively
    $C \in \Re^n \leftarrow$ mean, $A^{-1} \leftarrow$ covariance matrix, inverse chi-square t_p^1 and t_p^2
    Calculate mahalanobis distance to evaluate sample importance
    $M(X_t; C, A^{-1})$
    Obtain the highest two confidence candidate and take the average value from the output
    $y_1 \leftarrow highestCandidateOutput$
    $y_2 \leftarrow secondHighestCandidateOutput$
    $confidenceFinal \leftarrow (y_1 + y_2)/2$
    if $t_p^1 \leq M(X_t; C, A^{-1}) \leq t_p^2 \mid\mid confidenceFinal \leq \delta$ **then**
      Add a data sample X_t to the adaptive memory Dt
      $D_t = [D_{t-1}; X_t]$
    end if
  end if
end for

- *SP500 Problem*: This problem outlines time series prediction problem SP500 daily index from the period of January 3, 1950 till March 12, 2009. This problem consists of 14893 instances. The goal is to make one day ahead prediction using the previous four days indexes $y(t+1) = f(y(t), y(t-1), y(t-2), y(t-3), y(t-4))$. This problem reflects the volatile trait of stock data influenced by the dynamic market condition and is highly non-stationary especially due to the US economic recession happening around 2008 and 2009. This dataset is taken from Yahoo! finance website.

2 NADINE'S PSEUDOCODE

This section describes the learning policy of NADINE (algorithm 1) along with three sub algorithms of NADINE namely drift detection, correlation calculation, and adaptive memory mechanism which are described in the algorithm (2), (3), and (4) respectively.

REFERENCES

[1] P. Baldi, Paul D. Sadowski, and Daniel Whiteson. 2014. Searching for exotic particles in high-energy physics with deep learning. *Nature communications* 5 (2014), 4308.

[2] Andrea Coraddu, Luca Oneto, Alessandro Ghio, Stefano Savio, Davide Anguita, and Massimo Figari. 2014. Machine Learning Approaches for Improving Condition?Based Maintenance of Naval Propulsion Plants. *Journal of Engineering for the Maritime Environment* –, – (2014), –.

[3] G. Ditzler and R. Polikar. 2013. Incremental Learning of Concept Drift from Streaming Imbalanced Data. *IEEE Trans. on Knowl. and Data Eng.* 25, 10 (Oct. 2013), 2283–2301.

[4] R. Elwell and R. Polikar. 2011. Incremental Learning of Concept Drift in Nonstationary Environments. *Trans. Neur. Netw.* 22, 10 (Oct. 2011), 1517–1531.

[5] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. (2010). http://yann.lecun.com/exdb/mnist/

[6] David Lopez-Paz and Marc' Aurelio Ranzato. [n. d.]. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems 30*.

[7] Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. [n. d.]. Compete to Compute. In *Advances in Neural Information Processing Systems 26*.

[8] Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K. Chan. 2000. Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project. In *In Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*. IEEE Computer Press, 130–144.