NADINE's experimental results and dataset description

ACM Reference Format:
. 2019. NADINE's experimental results and dataset description. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 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 the details of NADINE's experimental results, dataset description, and the sub algorithms' pseudocode of NADINE learning policy namely drift detection, calculation of correlation, and adaptive memory mechanism.

1 NADINE'S EXPERIMENTAL RESULTS

This section provides the summary of NADINE's performance which is depicted in Table 1. For each dataset summarized in section 2, NADINE is tested five times and the highlighted average performance reflects the general performance of NADINE algorithm.

2 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.
- *RLCPS Problem*: the Record Linkage Comparison Patterns (RLCPS) problems consists of five million samples and twelve input features which represent patient particular information (e.g. first name, last name, etc). This dataset is taken from epidemicological cancer registry of the German state at North Rhine-Westphalia. This dataset was collected from the period of 2005 to 2008 and aims to determine whether the underlying record belong to one person [7]. In RLCPS, the imbalance data problem becomes the main dataset's characteristics where there are only 20931 out of 5 millions samples are categorized as matches (specific records belong to specific person), whereas the rest (most) of the records are the opposite (unmatches). In this experiement, we conduct the experiment under test-then-train prequential procedure with 5000 time stamps.
- *Indoor RFID Localization Problem*: the RFID localization is the four classes classification problem which aims to identify the object's location in the manufacturing shopfloor utilizing the signal from the RFID tag. From this tag, there are three features extracted from the signals captured by the RFID readers by placing the readers in different four zone manufacturing locations. Each location indicates the class of the sample. The dataset comprises 281.3 K data samples. The experiment is conducted under the test-then-train prequential procedure using 280 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 [8] is another benchmark problem of CL 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 [9] 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

Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1: Numerical results of NADINE algorithm for five experiments along with the average performance

Data Set	Trial	Classification rate (%)	Execution time(s)	Hidden Layers	Hidden Nodes	Number of parameters
SUSY	1	78.29 ± 0.02	1423	3.91 ± 1.15	113.2 ± 47.7	(3.26 ± 1.8)K
	2	77.98 ± 0.03	1107	2.37 ± 1.27	69.74 ± 55	(2.04 ± 2)K
	3	77.81 ± 0.03	2045	7.16 ± 3.77	182.77 ± 108.65	(4.84 ± 3.16)K
	4	77.89 ± 0.03	1399	3.89 ± 1.31	93.77 ± 42.61	(2.32 ± 1.34)K
	5	78.17 ± 0.03	1301	3.16 ± 1.87	99.94 ± 71.07	(2.87 ± 2.88)K
	Avg	**78.03 ± 0.03**	**1455**	**4.10 ± 1.87**	**111.88 ± 65.01**	**(3.07 ± 2.24)K**
HEPMASS 19%	1	83.67 ± 0.02	378	1.88 ± 1.07	28.42 ± 25.28	(0.603 ± 0.5)K
	2	84.11 ± 0.02	432	2.55 ± 1.37	49 ± 36.12	(1.06 ± 0.75)K
	3	83.84 ± 0.03	561	4.14 ± 2.08	82.9 ± 48.52	(1.59 ± 1.04)K
	4	83.94 ± 0.02	560	4.16 ± 1.62	89.53 ± 44.44	(2.12 ± 1.18)K
	5	83.53 ± 0.02	564	4.2 ± 2.41	89.63 ± 62.28	(2.11 ± 1.56)K
	Avg	**83.82 ± 0.02**	**499**	**3.39 ± 1.71**	**67.90 ± 43.33**	**(1.49 ± 1.01)K**
RLCPS	1	1 ± 0	978	1 ± 0	5.99 ± 0.07	74 ± 0
	2	99.99 ± 0.03	979	1 ± 0	12.99 ± 0.21	157.96 ± 1.48
	3	99.99 ± 0.03	982	1 ± 0	9.99 ± 0.13	121.99 ± 0.41
	4	99.99 ± 0.03	1027	1 ± 0	17.99 ± 0.28	217.96 ± 1.62
	5	99.99 ± 0.03	1039	1 ± 0	9.998 ± 0.13	122 ± 0.17
	Avg	**99.99 ± 0.02**	**1001**	**1 ± 0**	**11.39 ± 0.16**	**138.78 ± 0.74**
RFID localization	1	98.9 ± 3	51	1 ± 0	26.31 ± 4.69	215.25 ± 35.53
	2	98.9 ± 2	50	1 ± 0	24.36 ± 4.08	(199.59 ± 30.68
	3	98.32 ± 6.47	51	1 ± 0	24.78 ± 5.5	202.96 ± 42.58
	4	98.8 ± 3.73	53	1.16 ± 0.37	31.86 ± 14.71	385.56 ± 410.89
	5	98.53 ± 5.16	50	1 ± 0	22.25 ± 4.97	182.65 ± 38.51
	Avg	**98.69 ± 4.07**	**51**	**1.03 ± 0.07**	**25.91 ± 6.79**	**237.20 ± 111.64**
Mnist	1	87.41 ± 5.83	199	1 ± 0	23.14 ± 6.26	(18.67 ± 4.52)K
	2	85.8 ± 6.09	201	1 ± 0	16.56 ± 4.41	(13.36 ± 3.18)K
	3	85.7 ± 5.44	195	1.07 ± 0.26	18.24 ± 7.26	(13.66 ± 3.01)K
	4	86.3 ± 5.77	202	1 ± 0	16.14 ± 4.06	(13.02 ± 2.9)K
	5	88.19 ± 5.49	192	1 ± 0	26.55 ± 6.64	(21.42 ± 4.69)K
	Avg	**86.68 ± 5.72**	**198**	**1.01 ± 0.05**	**20.13 ± 5.73**	**(16.03 ± 3.66)K**
Rotated Mnist	1	75 ± 7.86	204	1 ± 0	18.05 ± 5.25	(14.57 ± 3.83)K
	2	74.9 ± 7.62	195	1 ± 0	16.59 ± 4.2	(13.4 ± 2.9)K
	3	73.85 ± 5.56	190	1 ± 0	10.97 ± 1.48	(8.86 ± 0.61)K
	4	75.6 ± 8.92	190	1 ± 0	20.12 ± 6.61	(16.25 ± 4.93)K
	5	73.22 ± 7.56	183	1 ± 0	12.72 ± 3.4	(10.27 ± 2.45)K
	Avg	**74.51 ± 7.50**	**192**	**1 ± 0**	**15.69 ± 4.188**	**(12.67 ± 2.944)K**
Permuted Mnist	1	76.28 ± 15.48	204	1.88 ± 0.32	42.55 ± 14.25	(9.88 ± 1.19)K
	2	79.61 ± 14.3	202	1 ± 0	19.72 ± 4.77	(15.9 ± 3.3)K
	3	79.41 ± 14.83	204	1 ± 0	20.14 ± 4.1	(16.24 ± 2.69)K
	4	76.75 ± 14.95	203	1.06 ± 0.24	15.82 ± 5.8	(11.95 ± 2.35)K
	5	76.18 ± 15.88	199	1 ± 0	13.43 ± 2.22	(10.84 ± 1.3)K
	Avg	**77.65 ± 15.09**	**202**	**1.19 ± 0.11**	**22.33 ± 6.23**	**(12.96 ± 2.17)K**
KDDCup 10%	1	99.84 ± 0.14	105	1 ± 0	11.93 ± 0.55	527.97 ± 10.8
	2	99.84 ± 0.15	98	1 ± 0	7.95 ± 0.37	352.58 ± 8.71
	3	99.84 ± 0.15	95	1 ± 0	11.91 ± 0.57	527.17 ± 13.08
	4	99.83 ± 0.17	95	1 ± 0	14.93 ± 0.66	660 ± 9.08
	5	99.85 ± 0.15	95	1 ± 0	14.91 ± 0.68	659.34 ± 12.82
	Avg	**99.84 ± 0.15**	**98**	**1 ± 0**	**12.33 ± 0.57**	**545.41 ± 10.90**
SEA	1	92.46 ± 6.03	15	1.11 ± 0.32	9.63 ± 8.49	76.75 ± 98.11
	2	92.26 ± 6.09	15	1.11 ± 0.32	10.91 ± 9.13	88.79 ± 114.15
	3	92.42 ± 6.17	14	1 ± 0	8.37 ± 2.83	52.69 ± 16.48
	4	91.77 ± 7.43	19	1.73 ± 0.65	26.99 ± 20.12	299.2 ± 333.01
	5	92.27 ± 6.26	14	1 ± 0	8.66 ± 3.15	54.41 ± 18.43
	Avg	**92.24 ± 6.40**	**15**	**1.19 ± 0.26**	**12.91 ± 8.74**	**114.37 ± 116.04**
Hyperplane	1	92.29 ± 3.08	16	1 ± 0	4.34 ± 0.86	32.61 ± 5.63
	2	92.22 ± 2.77	21	1.86 ± 0.35	23.21 ± 9.79	119.54 ± 46.84
	3	92.51 ± 2.02	16	1 ± 0	4.77 ± 0.66	35.64 ± 3.91
	4	92.32 ± 3.09	16	1 ± 0	3.87 ± 0.52	29.23 ± 3.14
	5	92.42 ± 2.51	16	1 ± 0	2.94 ± 0.29	22.76 ± 1.57
	Avg	**92.35 ± 2.69**	**17**	**1.17 ± 0.07**	**7.83 ± 2.42**	**47.95 ± 12.22**

of the various type of intrusions held in military network environment. This dataset was very popular dataset as it was used for many machine learning competitions (e.g. 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 changing 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.

- *Hyperplane Problem*: Hyperplane dataset is an artificial dataset generated by the massive online analysis (MOA) - a popular framework in the data stream field [2]. It is categorized as a binary classification problem which aims to separate data points into two classes based on to the position of d- dimensional random hyperplane $\sum_{j=1}^{d} w_j x_j > w_o$. This dataset has a gradual drift characteristic where the data samples are generated from one distribution with a probability of one. Then, this probability is reduced gradually until the second distribution completely replaces the first one. The hyperplane dataset consists of 120K sample and we conduct 120 time stamps for the experiment under prequential test-then-train procedure.

3 NADINE'S PSEUDOCODE

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

Algorithm 1 Drift Detection Method Based on Hoeffding Bound Concept

```
Input: Accuracy matrix $A \in \Re^{T\times n}$, $B \in \Re^{cut}$,$\epsilon_{drift}$,$\epsilon_{warning}$
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 2 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: $\underline{Scorr(i)}$ obtained from mean of mean HnCor-
            rOut(i): $\overline{\overline{HnCorrOut(i)}}$
        end for
        Obtain learningRate $\longleftarrow 0.01 * exp(-(1./Scorr - 1))$
    end if
end for
```

Algorithm 3 Adaptive Memory Storage Mechanism

```
Input: Data stream chunk $B_k \in \Re^{T\times n}$,$ca \longleftarrow n + 1$
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 recur-
        sively
        $C \in \Re^n \longleftarrow$ mean, $A^{-1} \longleftarrow$ covariance matrix, inverse chi-
        square $t_p^1$ and $t_p^2$
        Calculate mahalanobis distance to evaluate sample
        importance
        $M(X_t; C, A^{-1})$
        if $t_p^1 \leq M(X_t; C, A^{-1}) \leq t_p^2$ 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
```

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] Albert Bifet and Ricard GavaldÃă. 2007. Learning from time-changing data with adaptive windowing. In *In SIAM International Conference on Data Mining*.

[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. 2017. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 6467–6476. http://papers.nips.cc/paper/7225-gradient-episodic-memory-for-continual-learning.pdf

[7] Murat Sariyar, Andreas Borg, and Klaus Pommerening. 2011. Controlling False Match Rates in Record Linkage Using Extreme Value Theory. *J. of Biomedical Informatics* 44, 4 (Aug. 2011), 648–654. https://doi.org/10.1016/j.jbi.2011.02.008

[8] Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. 2013. Compete to Compute. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2310–2318. http://papers.nips.cc/paper/5059-compete-to-compute.pdf

[9] 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.