Supplementary Document

- **Paper Title:**
  A Skip-connected Evolving Recurrent Neural Network for Data Stream Classification under Label Latency Scenario
- **Authors:** Anonymous
- **Affiliation:** Anonymous
- **Comment:** The main manuscript has been submitted in AAAI 2020.

This document includes some additional information which could not be accommodated within the main manuscript due to the page limitation. The document consists of three annexures. Annexure-A presents the proposed SkipE-RNN based prediction in algorithmic format. Annexure-B includes the comparative results with respect to additional baselines and benchmark datasets, and finally, Annexure-C provides the results from ablation study with our proposed SkipE-RNN model.

Annexure-A

Algorithm 1: SkipE-RNN_based_Prediction(x,y,δ)

```
Input: $x = [x_1, \cdots, x_D]^\top$, Input to process
Input: $y = [y_1, \cdots, y_S]^\top$, Target output; $\{x^{(t)}, y^{(t)}\}$, $t = 1, \cdots, T$ is the stream
Input: $\delta$, Label latency
/* Assumption: Class label of at least one sample is available at the beginning.
/* Processing */
for $DC = 1$ to the no.of data chunk in the stream do
    for $t = 1$ to the no. of data sample in the chunk $DC$ do
        /*Testing Phase:
        Execute forward propagation considering $k = t - 1$; /* (ref. "Parameter Learning" Sec. in the Main paper) */
        Generate output class label $\widehat{y}^{(t)}$ /* (ref. "Parameter Learning" Sec. in the Main paper) */
        /*Training Phase:
        Perform hidden layer updating; /* ref. "Hidden Layer Adaptation" Sec. in the Main paper */
        if $y^{(t)}$ is available then
            $k^{(t)} = (t - 1)$; /* also assuming availability of initially labeled samples */
        end
        else
            $k^{(t)} = M(x^{(t)})$; /* ref. "Label Mapping Unit" Sec. in the Main paper */
        end
        Execute forward propagation;
        if $y^{(t-\delta)} \neq y^{(k^{(t-\delta)})}$ then
            Calculate loss function as per eq.(4) and eq. (5) [condition-1] and execute gradient computation; /* ref. "Regularized Weight Updating" Sec. in the Main paper */
            Update parameter as per eqs.(6)-(7) with $\psi = 1$; /* ref. "Regularized Weight Updating" Sec. in the Main paper*/
        end
        else
            Calculate loss function as per eq.(4) and eq. (5) [condition-2] and execute gradient computation; /*ref. "Regularized Weight Updating" Sec. in the Main paper */
            Update parameter as per eqs.(6)-(7) with $\psi = 0$; /* ref. "Regularized Weight Updating" Sec. in the Main paper */
        end
    end
end
```

The various symbols used in the Algorithm 1 conforms to the proposed SkipE-RNN model as described in the main manuscript. The flow of the algorithm can be better understood in combination with the Figure 1 in the main paper.

Annexure-B

The Annexure-B is comprised of two subsections, which include the additional information for the Case Study-1 and Case Study-2 in the main paper, respectively.

Case Study-1: Finite Latency Scenario

Datasets

Due to page limitation, though we mentioned about four datasets in the main paper, we actually evaluated SkipE-RNN with respect to six datasets: *i) Susy* [1], *ii) Electricity-pricing* [4], *iii) Hyperplane* [2], *iv) Sea* [13], *v) Rotated-MNIST* [8], and *vi) Permuted-MNIST* [8]. Susy is widely used as data stream for big data problem, whereas Electricity-pricing, Hyperplance, and Sea are well-used in literature as examples of data streams with variants of concept drifts (refer Table 1). The Electricity-pricing is a real-world data showing prominent temporal aspects. As clearly stated in literature [3], if the electricity price goes up now, “it is more likely than by chance to go up again, and vice versa”. This data is highly autocorrelated with “very clear cyclical peaks at every 24 hours, due to electricity consumption habits”. Though the dataset Sea is artificial/synthetic, it is prepared along with induced recurring environment [4], and consequently, this also shows temporal dependence. We use Permuted (P)-MNIST and Rotated (R)-MNIST as two variants of MNIST dataset where each task contains digits transformed by permutation of pixels or rotation by fixed angle $\in$[0°,180°], respectively. These two problems are popular continual learning problems and are put forward to evaluate against a high input dimension. The details of all these datasets are summarized in Table 1.

Table 1: Specifications for the datasets used in Case Study-1

Datasets	**Specifications**				
	#Instance	**#Attribute**	**#Target**	**#Task**	**Characteristics**
SUSY	5000000	18	2	5000	Synthetic, stationary
ELECT.	45312	8	2	45	Real, non-stationary with covariate drifts
HYPER.	120000	4	2	120	Synthetic, non-stationary with gradual concept drifts
SEA	100000	3	2	100	Synthetic, non-stationary with recurring concept drifts
R-MNIST	65000	784	10	65	Synthetic, non-stationary with abrupt concept drifts
P-MNIST	70000	784	10	70	Synthetic, non-stationary with recurring concept drifts

Baselines

Because of the same reason of page limitation, in the main manuscript, we included comparative study with only five baselines. However, originally, we evaluated SkipE-RNN in comparison with eight state-of-the-art models. All these algorithms deal with one or more of the various issues in concept drift scenario, including one-pass learning, learning of new knowledge, and preserving the previous knowledge (i.e. handling the issue of catastrophic forgetting) and so on. Thus, these are appropriate as baselines for streaming data analysis.

- **PNN** [11]: Progressive neural network; Primarily deals with concept evolution in a data stream;
- **DEN** [7]: Dynamically expandable network; Extension of PNN; Puts forward selective retraining, dynamic expansion, and splitting/duplicating methods.
- **HAT** [12]: Task-based hard attention mechanism; developed for concept evolution in a data stream;
- **pENsemble+** [10]: Built on the concept of evolving fuzzy parsimonious classifier; Equipped with online active learning and ensemble merging scenarios, which reduces operator annotation effort with reduced complexity.

- **Incremental Bagging** [9]: Online version of 'Bagging', an well-known ensemble learning model, working with low overhead. However, it requires large execution time.
- **Learn++.NSE** [4]: Appropriate for dealing with variants of drift scenarios; Capable of learning in non-stationary environments. However, it suffers from high structural complexity and considerably long execution time.
- **Online Multiclass (OMC) Boosting** [6]: A variant of online boosting algorithm. Based on ensemble learning; Uses optimal no. of classifier in the ensemble to achieve desired accuracy with reduced computational cost.
- **RNN_tanh**: Conventional RNN model [5] with single layer using *tanh* activation and learning based on back propagation through time (BPTT).

Incidentally, none of these above-mentioned baselines are designed for handling label latency. We, therefore, used these benchmarks to study the other issues in stream classification, as mentioned above, while keeping $\delta = 0$. Later, we further assess our model performance considering $\delta = 50$, $\delta = 100$, $\delta = 500$, and $\delta = 1000$.

Results and Discussions

The model performance is measured using four criteria, namely, *classification rate* (**CR**), *parameter count* (**PC**), *hidden unit count* (**HU**), and *execution time* (**ET**). The results of comparative study are presented through Table 2. The values are recorded as average of *five* random seeds. For I-bagging, OMC boosting, and Learn++.NSE, PC=NA and HU=NA (see Table 2), since these are based on decision tree model. Further, the I-bagging and Learn++.NSE models could not be executed on MNIST variants, because of the very high dimensionality of these data. On analyzing the results, we can infer the following.

Comparison on classification rate (CR): It is evident from the Table 2 that even with the constraint of single-pass data scanning and minimal usage of parameters, the proposed SkipE-RNN is able to achieve comparable and sometimes substantially better classification accuracy for each considered dataset. The average percentage improvement of SkipE-RNN over fixed structured RNN (using BPTT) model is more than 13%, and also, it is achieved with notably less execution time. The modelling of temporal dependencies through the dynamically adaptive recurrent architecture of SkipE-RNN leads the model to attain this encouraging performance. To be noted, the classification performance of SkipE-RNN for P-MNIST and R-MNIST datasets are significantly higher compared to the others. This further demonstrates the potentiality of the proposed SkipE-RNN in dealing with high dimensional and unstructured data.

Comparison on parameter count (PC) and hidden unit (HU) requirement: As shown in Table 2, though the number of parameters and hidden units for SkipE-RNN are sometimes higher (e.g. in case of P-MNIST and R-MNIST), the beauty lies in the fact that these hidden units and parameter requirements are not prefixed from the beginning. As depicted in the Figure 1, the execution of the proposed SkipE-RNN starts with only one unit in a single hidden layer, and then, gradually the number of hidden units are adjusted (added or removed) from the layer so as to cope up with the time varying distribution and conceptual drift of the streaming data. This gradual and on-the-fly structural adjustment helps SkipE-RNN in achieving desired accuracy with optimal number of parameters.

Comparison on Execution Time (ET): It may also be noted from Table 2 that even with the dynamic layer adaptation overhead, SkipE-RNN is able to achieve acceptable accuracy within reasonable time. Though online multiclass boosting (OMC) is in general popular for low computational time requirement, our proposed self-adaptive SkipE-RNN is found to be even faster than OMC-Boosting for each considered dataset. This is so because SkipE-RNN is based on single classifier and it works based on optimal no. of parameters.

Table 2: Comparative performance study of proposed SkipE-RNN under finite label latency $\delta = 0$

Data	**Models**	**Metrics**				
		CR	**HU**	**PC**	**ET**(s)	**Wilcoxon Test**
Susy	PNN	68.94 ± 4.08	60	424	345K	X
	DEN	63.15 ± 10.06	10	212	8K	X
	HAT	73.85 ± 3.18	20	342	16K	X
	OMC-Boosting	77.13 ± 1.43	NA	NA	14K	X
	Learn++.NSE	—	NA	NA	—	—
	I-Bagging	72.8 ± 3.1	NA	NA	73K	X
	pEnsemble+	76.99± 4.6	9	3249	35K	X
	RNN_tanh	64.31± 1.5	4	312	10K	X
	SkipE-RNN	**77.78 ± 1.96**	9	180	5K	
Electricity	PNN	57.84 ± 4.52	78	1868	51.345	X
	DEN	56.54 ± 7.66	16	178	72.54	X
	HAT	56.63 ± 8.04	20	242	145.57	X
	OMC-Boosting	77.12 ± 4.57	NA	NA	56.8	X
	Learn++.NSE	77.18 ± 9.3	NA	NA	169.88	X
	I-Bagging	72.8 ± 10.88	NA	NA	5K	X
	pEnsemble+	72.60± 12.1	3	243	78.2	X
	RNN_tanh	65.12± 7.21	4	104	35.41	X
	SkipE-RNN	**77.86 ± 5.3**	4	46	17.89	
Hyperplane	PNN	85.07 ± 7.12	42	560	190.196	X
	DEN	91.83 ± 4.17	8	58	202.57	X
	HAT	77.9 ± 10.76	12	98	370.8	X
	OMC-Boosting	86.18 ± 3.73	NA	NA	111.74	X
	Learn++.NSE	90.35 ± 2.48	NA	NA	374	X
	pEnsemble+	87.6 ± 6.2	3	75	150	X
	I-Bagging	81.39 ± 2.2	NA	NA	1.7K	X
	RNN_tanh	76.55 ± 2.82	4	80	101.58	X
	SkipE-RNN	**92.56 ± 2.15**	2	16	48.37	
Sea	PNN	84.87 ± 6.52	33	353	152.46	X
	DEN	79.95 ± 19.28	6	38	169.72	X
	HAT	74.65 ± 10.1	10	72	327	X
	OMC-Boosting	87.86 ± 3.85	NA	NA	77.44	X
	Learn++.NSE	90.17 ± 5.96	NA	NA	268	X
	I-Bagging	84.6 ± 13	NA	NA	1.5K	X
	pEnsemble+	92 ± 6	2	32	200	X
	RNN_tanh	75.17 ± 2.94	4	42	105.22	X
	SkipE-RNN	**90.63 ± 5.96**	2	14	48.92	
R-MNIST	PNN	56.19 ± 10.94	300	170K	128.91	X
	DEN	61.48 ± 21.75	440	290K	371.07	X
	HAT	64.52 ± 11.33	60	24.9K	190.59	X
	OMC-Boosting	26.07 ± 5.8	NA	NA	5K	X
	Learn++.NSE	NA	NA	NA	NA	—
	I-Bagging	NA	NA	NA	NA	—
	pEnsemble+	NA	NA	NA	NA	—
	RNN_tanh	63.23 ± 7.54	250	198K	207.49	X
	SkipE-RNN	**76.09 ± 5.65**	78	63K	92.42	
P-MNIST	PNN	64.42 ± 8.77	260	170K	152.95	X
	DEN	52.08 ± 22.6	440	290K	399.83	X
	HAT	59.64 ± 18.88	60	24.9K	207.04	X
	OMC-Boosting	35.58 ± 20.51	NA	NA	5K	X
	Learn++.NSE	NA	NA	NA	NA	—
	I-Bagging	NA	NA	NA	NA	—
	pEnsemble+	NA	NA	NA	NA	—
	RNN_tanh	69.62 ± 11.36	250	198K	425.10	X
	SkipE-RNN	**83.16 ± 13.42**	78	63K	198.75	

X: Reject the null hypothesis that a model performs better than SkipE-RNN
—: Main experimentation could not be conducted

Sensitivity on label latency: Figure 2 summarizes the performance of proposed SkipE-RNN-based prediction under various finite and non-zero latency scenario. It is evident from the figure that the classification performance of SkipE-RNN is not very sensitive to the latency/delay amount (δ). The average degradation in prediction accuracy is approximately 2.5% with respect to no-delay situation. This demonstrates the effectiveness of using the label mapping unit along with the recurrent feature learning, and also proves the utility of regularizing the model in case of label mismatch in subsequent phases. When the actual data label is not available, the model uses the label suggested by the mapping unit and dynamically configures the recurrent connection accordingly, with an earlier state of the network which is already updated with known label information, and thereby continues learning in delayed label scenario. Even if the suggested label is revealed to be incorrect, the model has the facility to penalize itself through regularization of its network parameters.

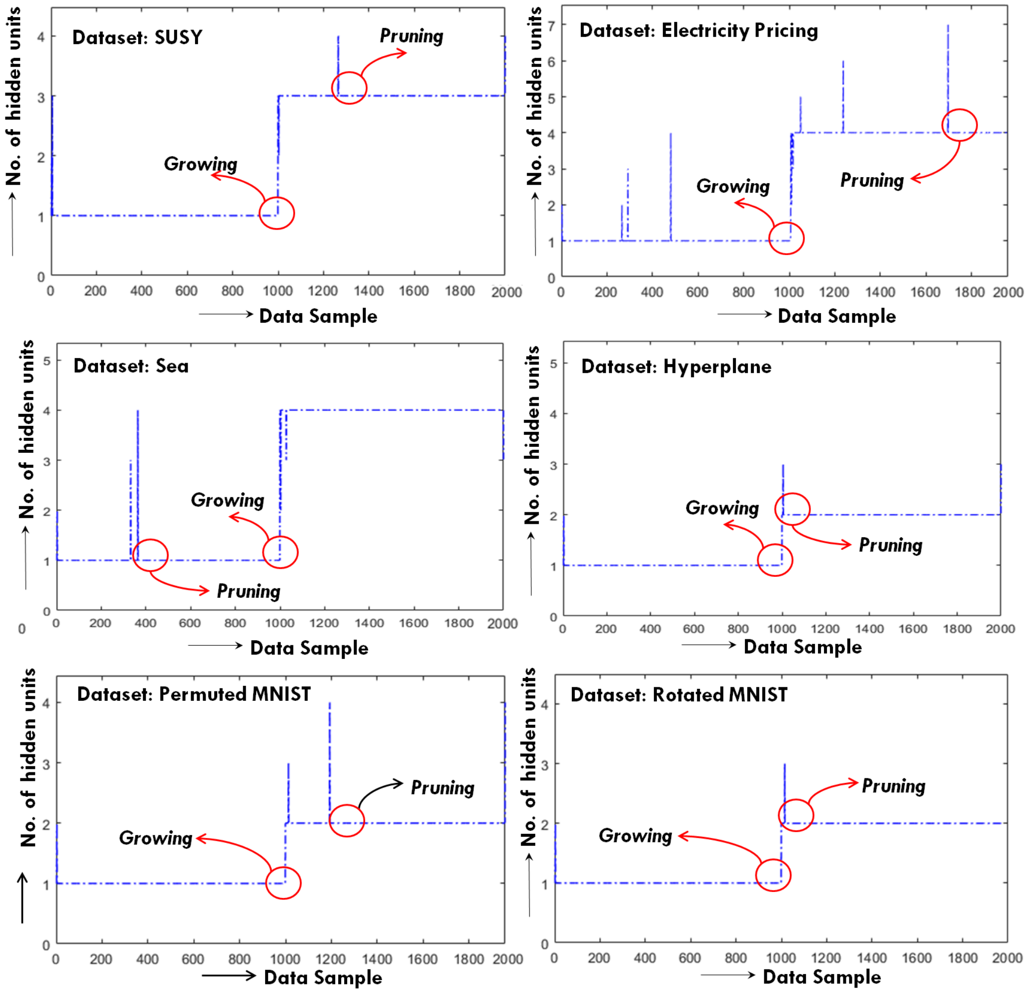

Figure 1: Hidden unit adjustment in SkipE-RNN for various datasets in case study-1 (see main paper)

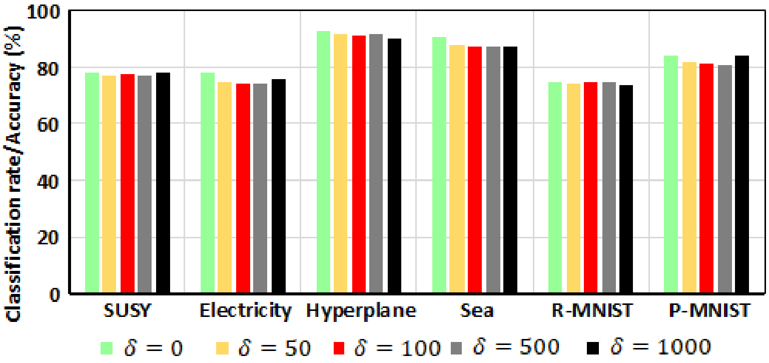

Figure 2: Comparative study of classification rate of SkipE-RNN considering different duration of finite latency (δ)

Results of statistical tests: The performance of SkipE-RNN is also validated using ***Wilcoxon statistical test***, as summarized in Table 2 (see right-most column). The results validate that, in every case, the proposed SkipE-RNN performs statistically better or similar, compared to the other considered models (with a significance level of 5%).

Case Study-2: Infinite Latency Scenario

The additional results of comparative study in terms of graphical plot of 'per step' classification performance is shown in the Figure 3 considering all the datasets used in Case Study-2 (refer to the main paper). It is evident from the figure that though SkipE-RNN sometimes starts with low accuracy at the very beginning, it has a powerful enough adaptation mechanism to finally reach the desired accuracy level. Judicious learning from initial samples through dynamically evolving skipped-recurrent-connection helps the model to achieve such classification performance in infinitely delayed data label scenario.

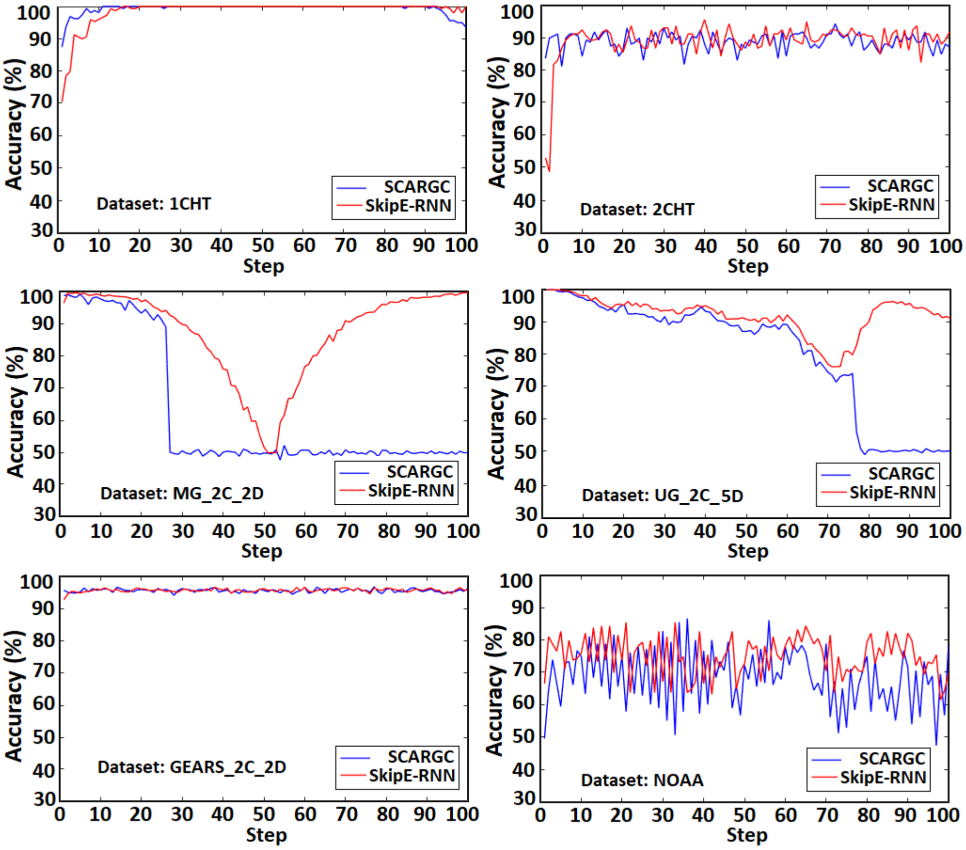

Figure 3: Comparative study of accuracy in the infinitely delayed label scenario (see case study-2 in main paper)

Annexure-C

In order to confirm the usefulness of autonomous structural evolution and regularized weight updating of SkipE-RNN, we have additionally performed ablation study considering *no pruning*, *no growing*, and *no regularization* features of the proposed SkipE-RNN (refer Table 3). As shown in the table, the study is conducted with respect to the datasets used in both the case studies in the main manuscript. The study evidently shows the performance deterioration of SkipE-RNN, either in terms of increased computational time (in case of no pruning) or in terms of decreased accuracy (especially in case of no growing and no regularization), without these model adaptation features during classification. The classification accuracy sometimes also deteriorates even when the number of hidden units is increased (e.g. in case of Hyperplane, Sea, R-MNIST etc.). This happens because of network over-fitting due to increased structural complexity. Further, it is also evident that when there is no regularization, the model performance is more affected for those datasets which shows higher temporal dependency, such as Electricity pricing, Sea, and Hyperplane. To be noted, in case of 1CHT, 2CHT, MG-2C-2D, UG-2C-5D, GEARS-2C-2D, and NOAA datasets, the scheme of using no regularization is equivalent to the infinite latency scenario. Thus, the results remain the same as that reported in the Case study-2 (see the Main manuscript), and so, we do not repeat this here.

Table 3: Summary of the ablation study for SkipE-RNN

Ablation Scheme	**Datasets**	**Metrics**			
		CR	**HU**	**PC**	**ET** (sec.)
Proposed Model WITHOUT hidden unit pruning feature	Susy	76.38 ±03.83	15	317	6.25K
	Electricity	76.50 ± 08.52	7	79	22.57
	Hyperplane	89.53 ± 02.77	3	23	55.63
	Sea	87.14 ± 14.22	3	20	51.50
	R-MNIST	73.49 ± 05.06	158	126K	1.08K
	P-MNIST	83.09 ± 13.37	157	125K	1.09K
	1CHT	99.53 ± 1.15	3	17	11.72
	2CHT	85.94 ± 6.38	3	17	11.48
	GEARS_2C_2D	95.69 ± 0.80	3	17	93.11
	MG_2C_2D	88.05 ± 11.65	3	17	96.89
	UG_2C_5D	92.38 ± 5.48	4	34	98.09
	NOAA	74.87 ± 3.38	6	68	11.39
Proposed Model WITHOUT hidden unit growing feature	Susy	76.55 ± 1.33	1	23	2.25K
	Electricity	75.21 ± 9.82	1	13	16.61
	Hyperplane	89.90 ± 6.14	1	9	42.43
	Sea	83.21 ± 9.41	1	8	36.93
	R-MNIST	19.69 ± 2.02	1	805	87.01
	P-MNIST	18.96 ± 1.89	1	805	180.74
	1CHT	99.50 ± 1.26	1	7	06.89
	2CHT	87.07 ± 3.30	1	7	06.09
	GEARS_2C_2D	95.5 ± 1.46	1	7	89.93
	MG_2C_2D	87.18 ± 14.10	1	7	89.25
	UG_2C_5D	80.73 ± 16.44	1	10	81.78
	NOAA	71.41 ± 3.39	1	13	07.81
Proposed Model WITHOUT regularization	Susy	77.54 ± 1.46	9	180	2.8K
	Electricity	74.88 ± 10.95	4	46	17.6
	Hyperplane	87.50 ± 2.45	2	16	47.21
	Sea	86.82 ± 7.19	2	14	44.64
	R-MNIST	75.95 ± 04.90	78	63K	91.66
	P-MNIST	83.02 ± 13.78	78	63K	196.5

References

[1] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5:4308, 2014.

[2] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *Journal of Machine Learning Research*, 11(May):1601–1604, 2010.

[3] Albert Bifet, Jesse Read, Indrė Žliobaitė, Bernhard Pfahringer, and Geoff Holmes. Pitfalls in benchmarking data stream classification and how to avoid them. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 465–479, 2013.

[4] Gregory Ditzler and Robi Polikar. Incremental learning of concept drift from streaming imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2283–2301, 2013.

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[6] Young Hun Jung, Jack Goetz, and Ambuj Tewari. Online multiclass boosting. In *Thirty-First conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[7] Jeongtae Lee, Jaehong Yun, Sungju Hwang, and Eunho Yang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

[8] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6467–6476, 2017.

[9] C Oza Nikunj and January Russell Stuart. Online bagging and boosting. jaakkola tommi and richardson thomas, editors. In *Eighth International Workshop on Artificial Intelligence and Statistics*, pages 105–112, 2001.

[10] Mahardhika Pratama, Eric Dimla, Tegoeh Tjahjowidodo, Witold Pedrycz, and Edwin Lughofer. Online tool condition monitoring based on parsimonious ensemble+. *IEEE Transactions on Cybernetics*, 2018.

[11] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[12] Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *35th International Conference on Machine Learning*, pages 4548–4557, 2018.

[13] W Nick Street and YongSeog Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382. ACM, 2001.