Augmented Memory replay based Continual Learning Approaches for Network Intrusion Detection

S.K. Amalapuram, S.S. Channappayya, B.R. Tamma

Networked Wireless Systems (NeWS) lab,
Department of Computer Science and Engineering,
Indian Institute of Technology Hyderabad, India

37^{th} conference on Neural Information Processing System,
New Orleans, USA

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

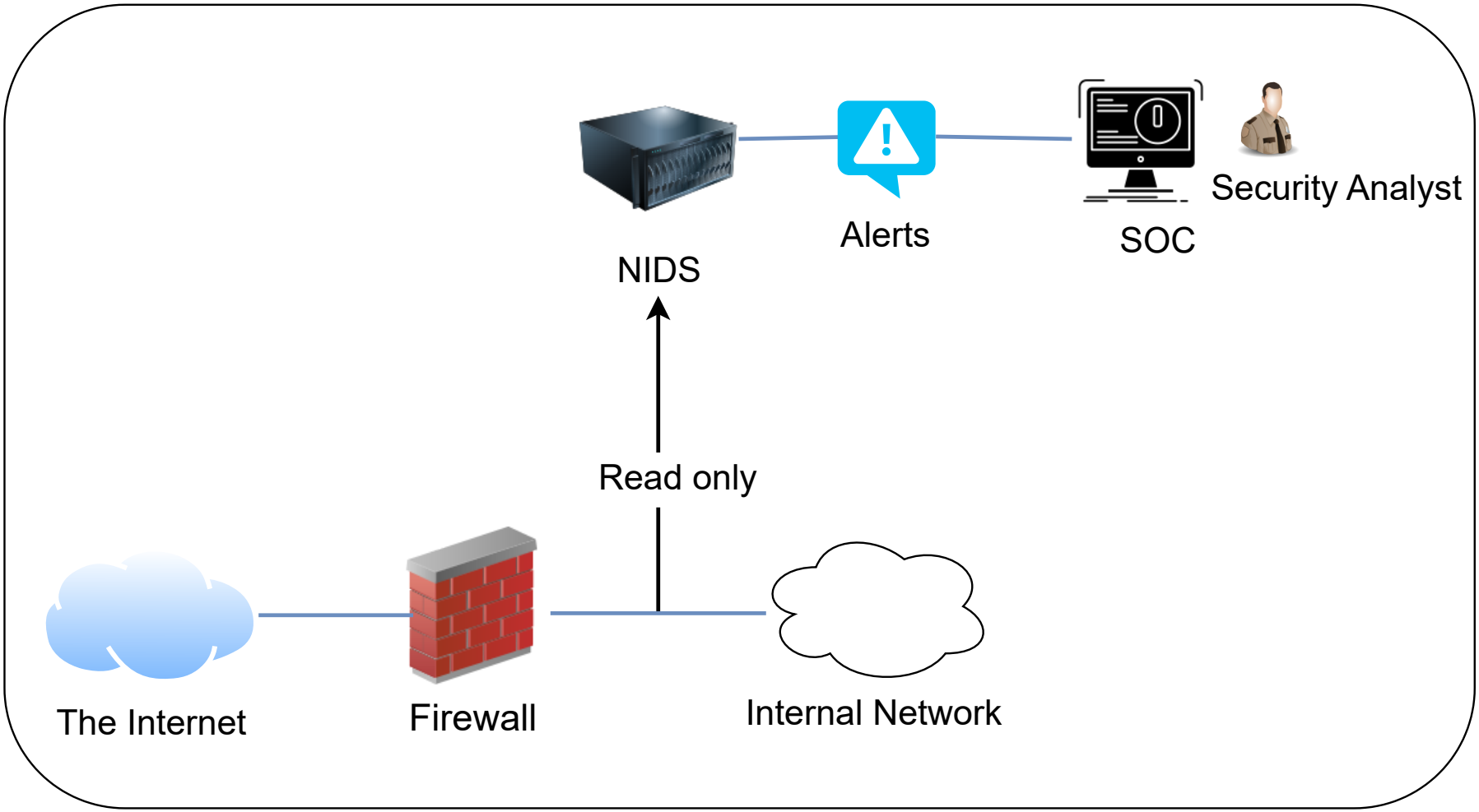


Figure 1: An operational position of NIDS

Motivation (severe class imbalance in datasets)[1][2]

CICIDS-2017

- Candian Institute for Cybersecurity
- No of samples: 2.1 million
- Heavy Class Imbalance (CI)- Benign traffic (84%)

ANOSHIFT subset

- No of samples: 11 million
- Heavy Class Imbalance (CI)- Attack traffic (90%)

CICIDS-2018

- No of samples: 63 million
- Heavy Class Imbalance (CI)- Benign traffic (84%)

[1] Liu Lisa et al. "Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018". In: *2022 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2022, pp. 254–262.

[2] Marius Dragoi et al. "AnoShift: A distribution shift benchmark for unsupervised anomaly detection". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 32854–32867.

Severe class imbalance (SCI)

- Diff. bw #samples among minority classes is **high** ?
- How much **high** ?

[3] Aristotelis Chrysakis and Marie-Francine Moens. "Online continual learning from imbalanced data". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1952–1961.

Severe class imbalance (SCI)

- Diff. bw #samples among minority classes is **high** ?
- How much **high** ?
- **2.1 million** between DDOS attack-HOIC and SQL Injection minority classes of the CICIDS-2018 dataset

[3] Aristotelis Chrysakis and Marie-Francine Moens. "Online continual learning from imbalanced data". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1952–1961.

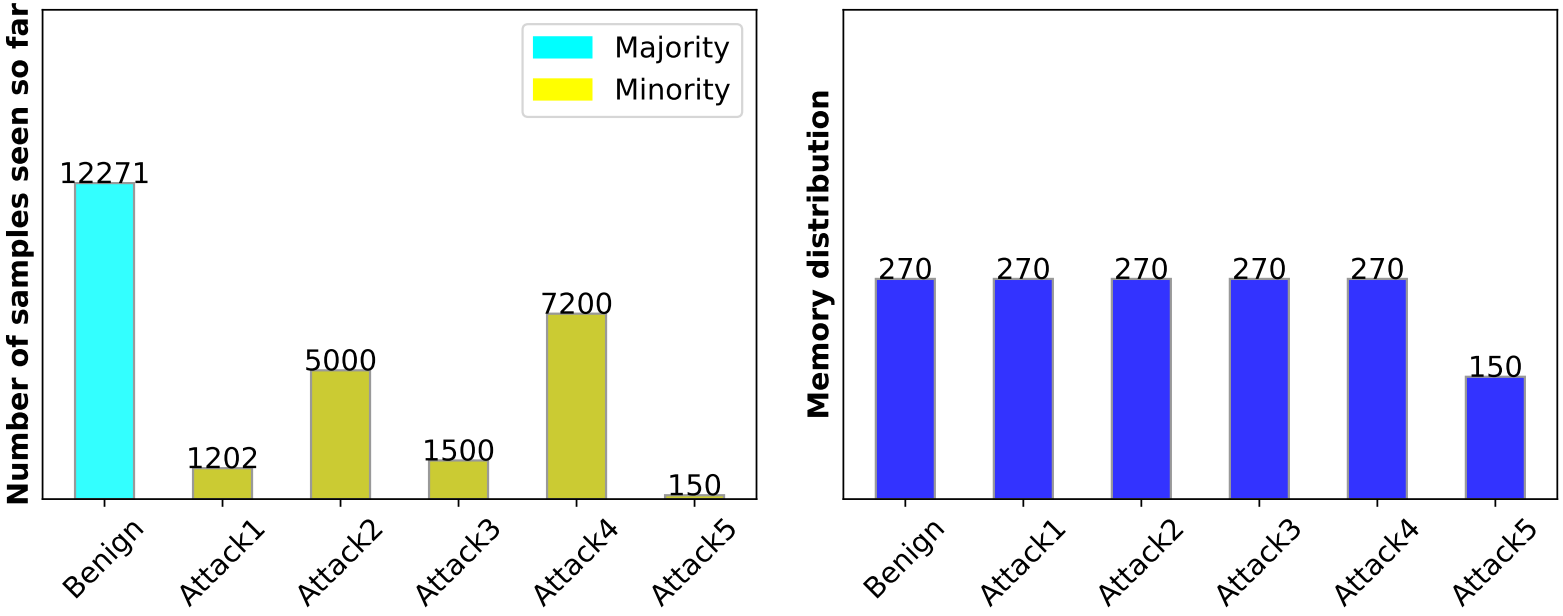
Severe class imbalance (SCI)

- Diff. bw #samples among minority classes is **high** ?
- How much **high** ?
- **2.1 million** between DDOS attack-HOIC and SQL Injection minority classes of the CICIDS-2018 dataset

Class balanced reservoir sampling (CBRS)

- Under SCI, treats different class samples equally in the buffer memory
- Due to reliance on local information
- Conflict of equal weights

[3] Aristotelis Chrysakis and Marie-Francine Moens. "Online continual learning from imbalanced data". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1952–1961.

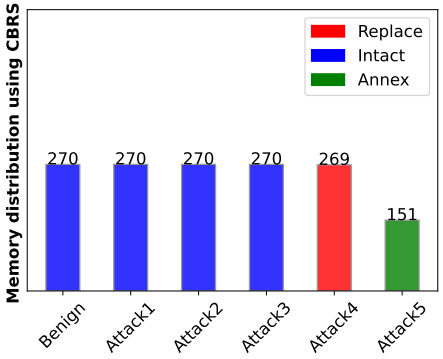


(a) Running statistics

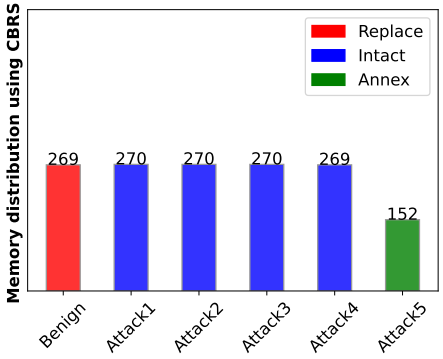
(b) Memory distribution

Figure 2: Comparison between CBRS and ECBRS CICIDS-2018 dataset with $(\mathbf{M}) = 1500$. (a) Running statistics indicate the number of classwise samples seen so far. (b) Memory distribution represents the strength of each class in buffer memory at a particular instance.

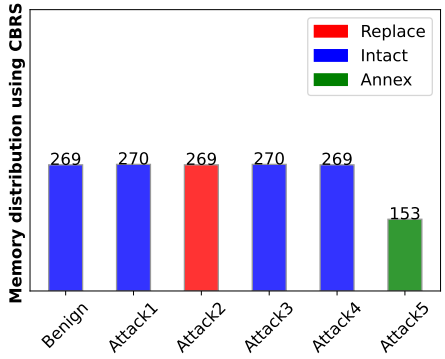
Illustration of variations in the CBRS buffer memory configuration



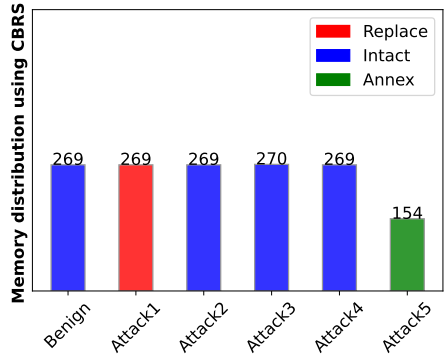
(a)



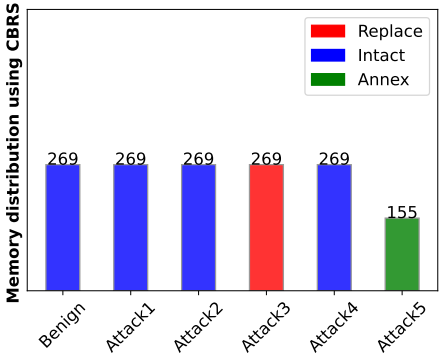
(b)



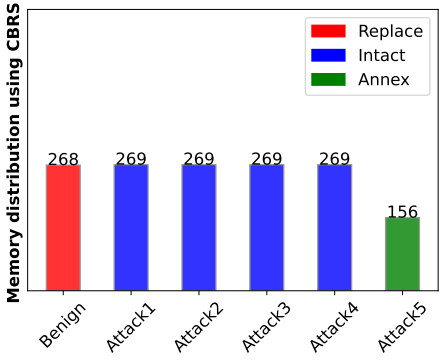
(c)



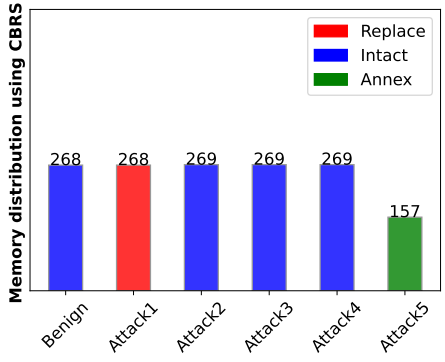
(d)



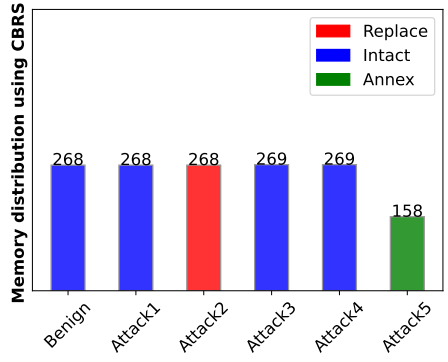
(e)



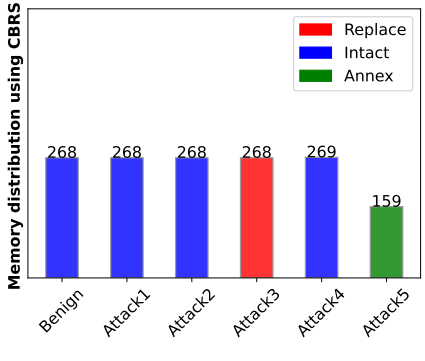
(f)



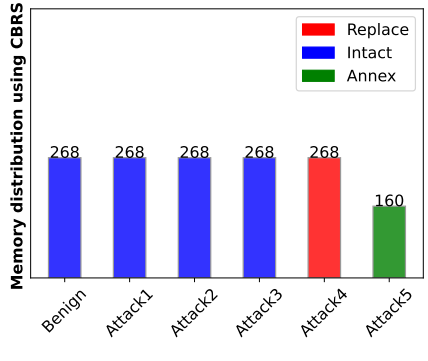
(g)



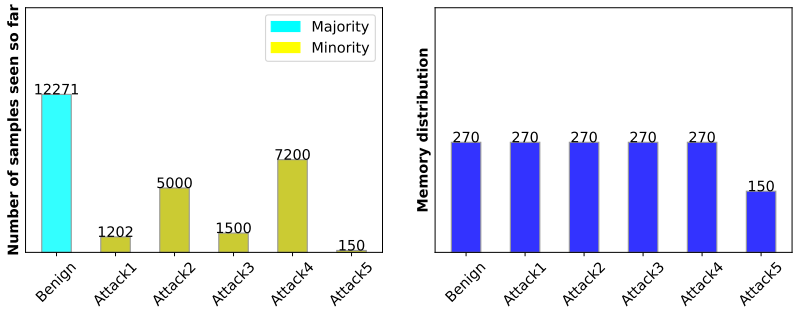
(h)



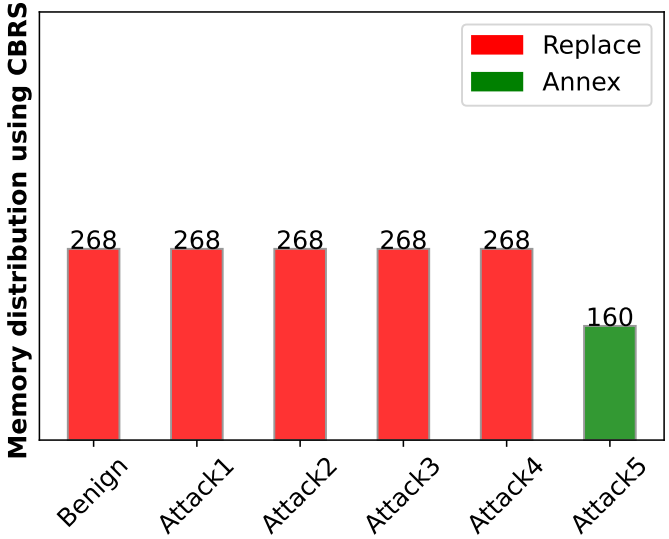
(i)



(j)



(a) Running statistics (b) Memory distribution



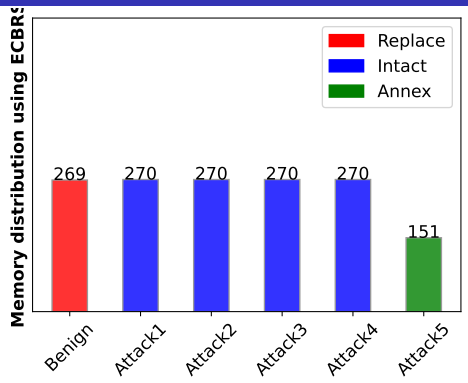
(c) CBRS

Figure 4: Comparison between CBRS and ECBRS CICIDS-2018 dataset with $(\mathbf{M}) = 1500$. (a) Running statistics indicate the number of classwise samples seen so far. (b) Memory distribution represents the strength of each class in buffer memory at a particular instance. Upon the arrival of ten new samples from the class *attack5*, using CBRS, the memory distribution changes to (c). For (c), the class with a red-colored bar is chosen for replacement, the green-colored bar class receives new samples, and the class with the blue-colored bar remains intact.

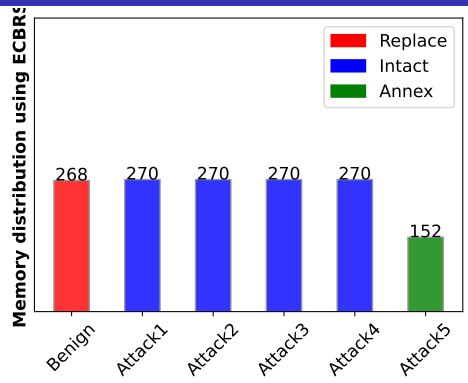
Input: data stream: $(x_i, y_i)_{i=1}^{n}$, number of currently stored instances of class (c$\equiv y_i$): m_c, number of stream instances of class c $\equiv y_i$ encountered thus far: n_c

for $i = 1$ **to** n **do**
  if memory is **not** filled **then**
    store (x_i, y_i)
  else
    if y_i is **not** a full class **then**
      select the largest class with higher running statistics value and non-zero samples with $m_c \geq \gamma(c)$ in the buffer. Otherwise, select class with the next higher statistic value
      overwrite the selected class sample with (x_i, y_i)
    else
      sample u $\sim$ Uniform(0,1)
      if u $\leq m_c/n_c$ **then**
        pick a stored instance of class c $\equiv y_i$ at random and replace it with (x_i, y_i)
      else
        Ignore (x_i, y_i)
      end if
    end if
  end if
end for

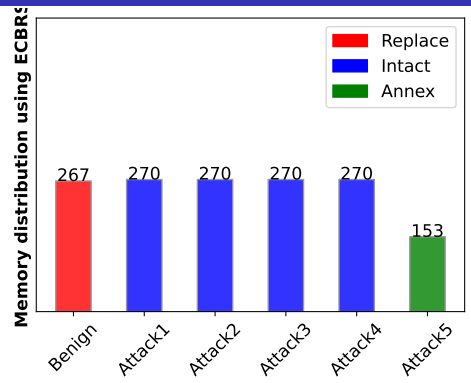
Illustration of variations in the ECBRS buffer memory configuration



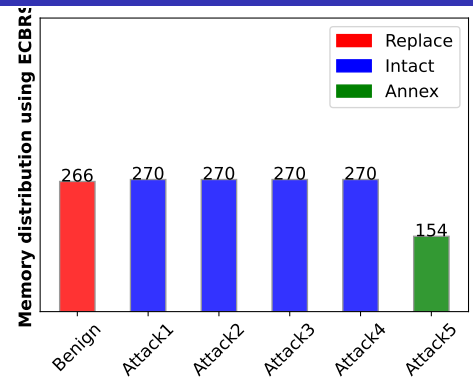
(a)



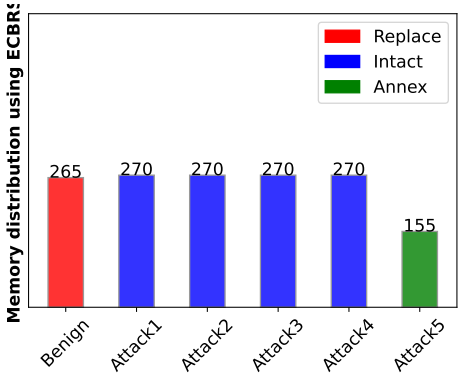
(b)



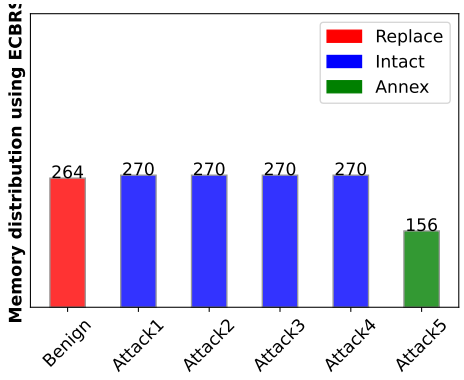
(c)



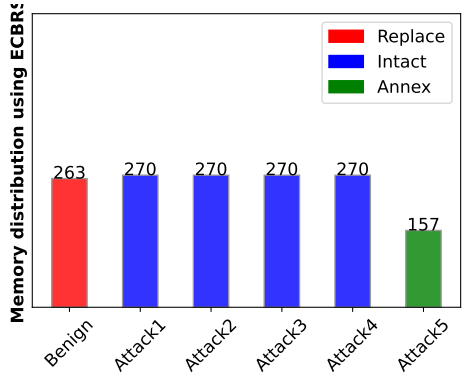
(d)



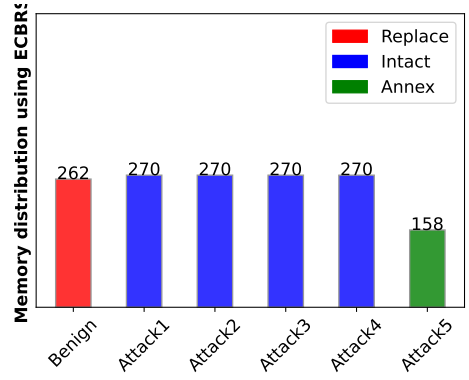
(e)



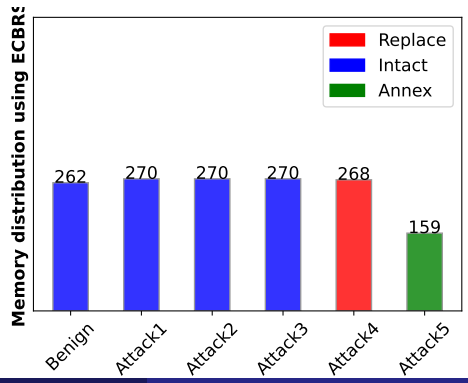
(f)

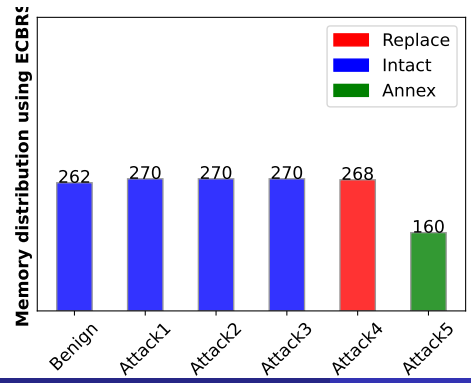


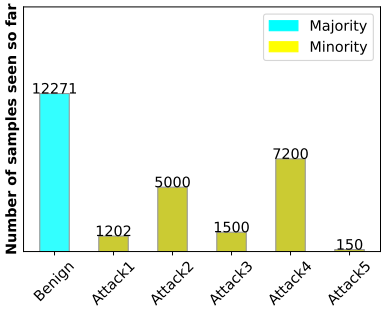
(g)



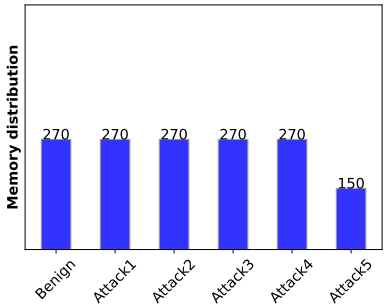
(h)



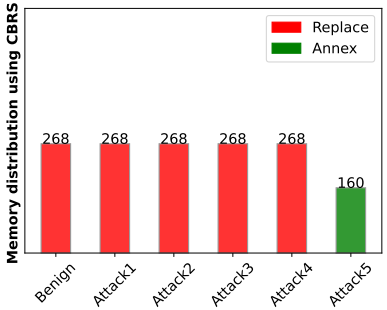




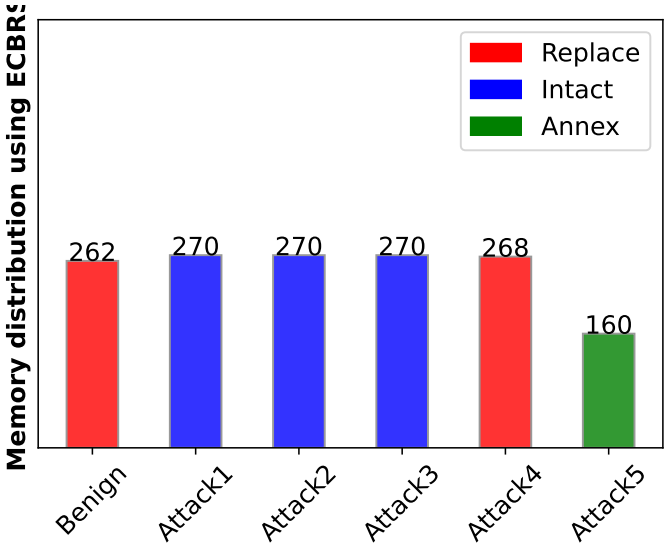
(a) Running statistics



(b) Memory distribution



(c) CBRS



(d) ECBRS

Figure 6: Comparison between CBRS and ECBRS CICIDS-2018 dataset with $(\mathbf{M}) = 1500$. (a) Running statistics indicate the number of classwise samples seen so far. (b) Memory distribution represents the strength of each class in buffer memory at a particular instance. Upon the arrival of ten new samples from the class *attack5*, using ECBRS, the memory distribution changes to (d). For (c) and (d), the class with a red-colored bar is chosen for replacement, the green-colored bar class receives new samples, and the class with the blue-colored bar remains intact.

Table 1: Performance results comparison of the proposed ECBRS method with the baselines with each experiment repeated five times independently. The performance result of the MIR using ECBRS as a memory population method is highlighted in light grey color. We observe the improved performance over the MIR with the random memory population method.

	CICIDS-2017			CICIDS-2018			UNSW-NB15			CTU-13		
Baseline Methods	PR-AUC (A)	PR-AUC (B)	ROC-AUC	PR-AUC (A)	PR-AUC (B)	ROC-AUC	PR-AUC (A)	PR-AUC (B)	ROC-AUC	PR-AUC (A)	PR-AUC (B)	ROC-AUC
EWC	0.617	0.766	0.608	0.740	0.762	0.505	0.925	0.823	0.913	1.00	1.00	0.999
SI	0.812	0.878	0.868	0.804	0.826	0.744	0.985	0.989	0.990	1.00	1.00	0.999
GEM	0.993	0.988	0.991	0.739	0.762	0.502	0.998	0.995	0.994	1.00	1.00	0.999
A-GEM	0.84	0.852	0.696	0.738	0.762	0.5	0.750	0.750	0.500	0.750	0.750	0.500
GSS-greedy	0.807	0.827	0.742	0.8215	0.762	0.664	0.949	0.848	0.959	1.00	1.00	0.999
MIR	0.785	0.840	0.798	0.737	0.762	0.5	0.886	0.807	0.855	0.950	0.950	0.899
CBRS	0.999	0.999	0.999	0.999	0.999	0.998	**0.999**	**0.999**	**0.999**	1.00	0.999	0.999
ECBRS (ours)	**1.00**	**0.999**	**0.999**	**0.999**	**0.999**	**0.998**	**0.999**	**0.999**	**0.999**	**1.00**	**1.00**	**0.999**
MIR + ECBRS	1.00	1.00	0.999	0.994	0.994	0.992	0.999	0.999	0.999	1.00	1.00	0.999

	ANOSHIFT			CIFAR-100			CLEAR-10			CLEAR-100		
Baseline Methods	PR-AUC (A)	PR-AUC (B)	ROC-AUC	PR-AUC (A)	PR-AUC (B)	ROC-AUC	PR-AUC (A)	PR-AUC (B)	ROC-AUC	PR-AUC (A)	PR-AUC (B)	ROC-AUC
EWC	0.543	0.566	0.550	0.644	0.603	0.644	0.858	0.828	0.835	0.770	0.764	0.778
SI	0.620	0.609	0.631	0.645	0.614	0.635	0.847	0.833	0.835	0.766	0.753	0.772
GEM	0.880	0.900	0.902	0.653	0.626	0.643	0.858	0.852	0.848	0.818	0.800	0.813
A-GEM	0.846	0.900	0.883	0.638	0.591	0.638	0.853	0.822	0.833	0.772	0.766	0.780
GSS-greedy	0.742	0.744	0.753	0.659	0.646	0.662	0.881	0.850	0.861	0.733	0.691	0.721
MIR	0.655	0.609	0.620	0.640	0.640	0.636	0.890	0.885	0.887	0.837	0.800	0.820
CBRS	0.949	0.939	0.941	0.572	0.598	0.572	0.941	0.927	0.931	0.841	0.789	0.820
ECBRS (ours)	**0.949**	**0.944**	**0.948**	0.663	0.611	0.663	0.937	0.926	0.926	**0.854**	**0.807**	**0.831**
MIR + ECBRS	0.942	0.928	0.934	**0.663**	**0.659**	**0.663**	**0.953**	**0.933**	**0.942**	0.839	0.793	0.817

Independent ECBRS module

- Onpar/outperform on
  all datasets
- ECBRS outperforms CBRS when
  1. DS in benign and attack
  2. ANOSHIFT
  3. SVHN, CIFAR-10/100, and CLEAR-10/100

Independent ECBRS module

- Onpar/outperform on all datasets
- ECBRS outperforms CBRS when
  1. DS in benign and attack
  2. ANOSHIFT
  3. SVHN, CIFAR-10/100, and CLEAR-10/100
- ECBRS outperforms on CBRS (aforementioned datasets)
  1. On avg. 7% on attack data
  2. On avg. 3% on benign data

Independent ECBRS module

- Onpar/outperform on
  all datasets
- ECBRS outperforms CBRS when
  1. DS in benign and attack
  2. ANOSHIFT
  3. SVHN, CIFAR-10/100, and CLEAR-10/100
- ECBRS outperforms on CBRS (aforementioned datasets)
  1. On avg. 7% on attack data
  2. On avg. 3% on benign data
- GEM, A-GEM are competitive
  1. Ring buffer memory policy
  2. Positive backaward transfer

Independent ECBRS module

- Onpar/outperform on all datasets
- ECBRS outperforms CBRS when
  1. DS in benign and attack
  2. ANOSHIFT
  3. SVHN, CIFAR-10/100, and CLEAR-10/100
- ECBRS outperforms on CBRS (aforementioned datasets)
  1. On avg. 7% on attack data
  2. On avg. 3% on benign data
- GEM, A-GEM are competitive
  1. Ring buffer memory policy
  2. Positive backaward transfer

ECBRS as **memory population** module

- Experimented with MIR
  1. MIR+ECBRS
- MIR+ECBRS outperforms MIR when

Independent ECBRS module

- Onpar/outperform on all datasets
- ECBRS outperforms CBRS when
  1. DS in benign and attack
  2. ANOSHIFT
  3. SVHN, CIFAR-10/100, and CLEAR-10/100
- ECBRS outperforms on CBRS (aforementioned datasets)
  1. On avg. 7% on attack data
  2. On avg. 3% on benign data
- GEM, A-GEM are competitive
  1. Ring buffer memory policy
  2. Positive backaward transfer

ECBRS as **memory population** module

- Experimented with MIR
  1. MIR+ECBRS
- MIR+ECBRS outperforms MIR when
  1. On all datasets in benign and attack
  2. On avg. 12% on attack data
  3. On avg. 14% ob benign data
- ECBRS gain high on large-scale datasets (LSD)

Independent ECBRS module

- Onpar/outperform on all datasets
- ECBRS outperforms CBRS when
  1. DS in benign and attack
  2. ANOSHIFT
  3. SVHN, CIFAR-10/100, and CLEAR-10/100
- ECBRS outperforms on CBRS (aforementioned datasets)
  1. On avg. 7% on attack data
  2. On avg. 3% on benign data
- GEM, A-GEM are competitive
  1. Ring buffer memory policy
  2. Positive backaward transfer

ECBRS as **memory population** module

- Experimented with MIR
  1. MIR+ECBRS
- MIR+ECBRS outperforms MIR when
  1. On all datasets in benign and attack
  2. On avg. 12% on attack data
  3. On avg. 14% ob benign data
- ECBRS gain high on large-scale datasets (LSD)
  1. ANOSHIFT,CICIDS-2017/2018, and UNSW-NB15
  2. On avg. 30% on attack data
  3. On avg. 31% ob benign data

Maximally interfered retrieval (MIR)

- Computes virtual SGD updates
- Let current model (Θ_r)
  1. Train (Θ_r) with the current batch $(\mathbf{B})$ of samples $(\Theta_r \Rightarrow \Theta_v)$
  2. Use Θ_v to find interfering samples from $(\mathbf{M})$
  3. $\ell_r = \ell(\mathbf{M}, \Theta_r)$ and $\ell_v = \ell(\mathbf{M}, \Theta_v)$
  4. Interfered samples ($\ell_v > \ell_r$)
  5. Train Θ_r using $\mathbf{B}$ and $\mathbf{M}_i$
  6. Ignore Θ_v
- Θ_v has a performance hit on LSD

[4] Aljundi Rahaf and Caccia Lucas. "Online Continual Learning with Maximally Interfered Retrieval". In: *NIPS*. 2019.

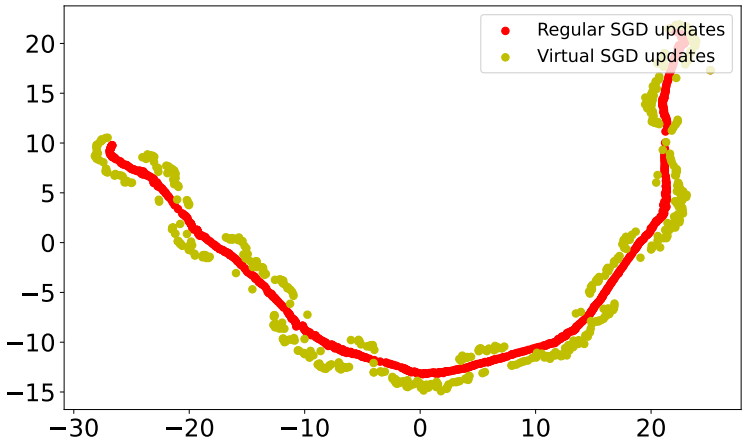
Maximally interfered retrieval (MIR)

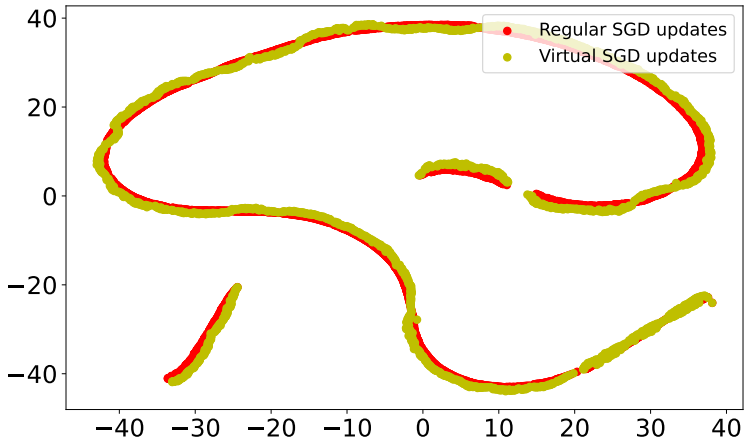
- Computes virtual SGD updates
- Let current model (Θ_r)
  1. Train (Θ_r) with the current batch $(\mathbf{B})$ of samples $(\Theta_r \Rightarrow \Theta_v)$
  2. Use Θ_v to find interfering samples from $(\mathbf{M})$
  3. $\ell_r = \ell(\mathbf{M}, \Theta_r)$ and $\ell_v = \ell(\mathbf{M}, \Theta_v)$
  4. Interfered samples ($\ell_v > \ell_r$)
  5. Train Θ_r using $\mathbf{B}$ and $\mathbf{M}_i$
  6. Ignore Θ_v
- Θ_v has a performance hit on LSD

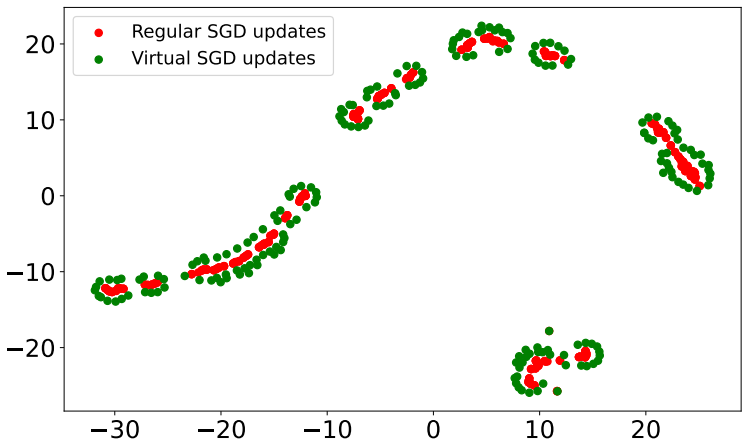
Table 2: Timing comparison between the virtual SGD operations and the total training of the MIR algorithm.

Dataset	Θ_v time	Train time	Proportion time
CICIDS-2017	102.9	254.3	59.5%
UNSW-NB15	132.4	428.2	30.9 %
CTU-13	142.4	398.1	35.7%
KDDCUP'99	191.0	420.8	45.3%
ANOSHIFT	499.4	1210.4	41.2%
CICIDS-2018	33315.0	7620	43.5%
SVHN	112.0	217.0	51.6%
CIFAR-10	58.6	118.6	49.4%
CIFAR-100	45.71	88.56	48.38%
CLEAR-10	128.2	265.9	51.7%
CLEAR-100	703.6	1490	47.2%

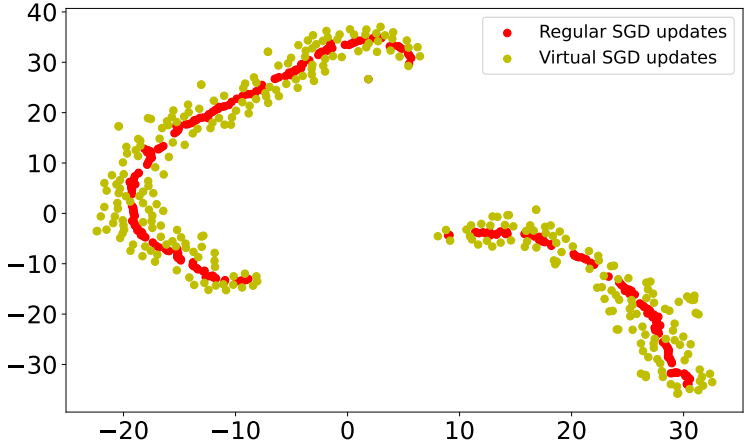
[4] Aljundi Rahaf and Caccia Lucas. "Online Continual Learning with Maximally Interfered Retrieval". In *NIPS*. 2019.







(c) CIFAR-10



(d) CLEAR-10

Figure 7: t-SNE visualization of regular SGD updates and virtual SGD updates. An MLP is trained on the CICIDS-2017 and CICIDS-2018 datasets. A ResNet-18 (pretrain) model is trained on the CIFAR-10 and CLEAR-10 datasets.

Formalizing the observations

- Relation bw Θ_r and Θ_v
  1. Overlaps
  2. Scattered around
- $\Theta_v = \varepsilon + \Theta_r$

Formalizing the observations

- Relation bw Θ_r and Θ_v
  1. Overlaps
  2. Scattered around
- $\Theta_v = \varepsilon + \Theta_r$

How to estimate ε ?

- Understanding error $(\mathbf{E})$ distribution $(\Theta_v - \Theta_r)$
- Modelling the error to estimate ε

The error distribution between the Θ_r and Θ_v

Formalizing the observations

- Relation bw Θ_r and Θ_v
  1. Overlaps
  2. Scattered around
- $\Theta_v = \varepsilon + \Theta_r$

How to estimate ε ?

- Understanding error $(\mathbf{E})$ distribution $(\Theta_v - \Theta_r)$
- Modelling the error to estimate ε

The error distribution between the Θ_r and Θ_v

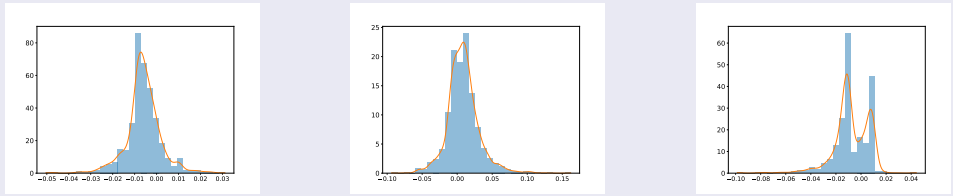


Figure 8: CICIDS-2017

Formalizing the observations

- Relation bw Θ_r and Θ_v
  1. Overlaps
  2. Scattered around
- $\Theta_v = \varepsilon + \Theta_r$

How to estimate ε ?

- Understanding error $(\mathbf{E})$ distribution $(\Theta_v - \Theta_r)$
- Modelling the error to estimate ε

The error distribution between the Θ_r and Θ_v

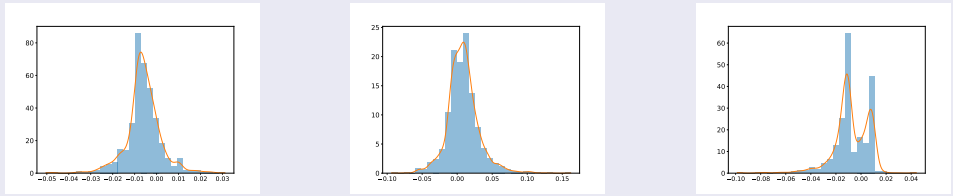


Figure 8: CICIDS-2017

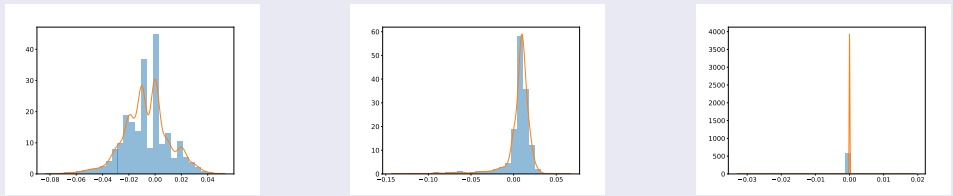


Figure 9: CICIDS-2018

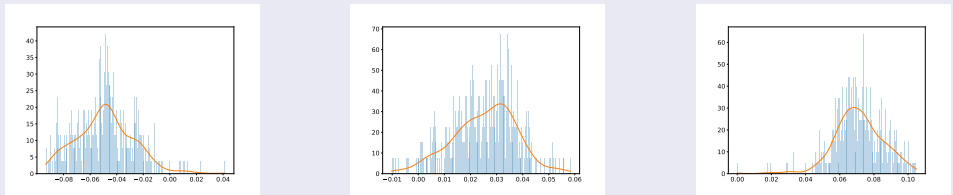


Figure 10: CIFAR-10

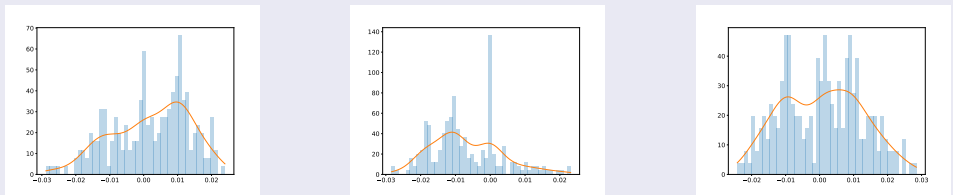


Figure 11: CLEAR-10

Modeling

- Gaussian mixture model
- z $\sim$ GMM
- $\Theta_v = z + \Theta_r$

Train GMM ?

- Train first task with MIR
- Compute Θ_v and Θ_r
- Compute $\mathbf{E}$
- Train GMM using $\mathbf{E}$

PAPA's way to compute interferecing samples

- Computes virtual SGD updates
- Let current model (Θ_r)
  1. $\mathbf{Z} \sim GMM$
  2. Compute $\Theta_v = \mathbf{Z} + \Theta_r$
  3. Use Θ_v to find interfering samples from ($\mathbf{M}$)
  4. $\ell_r = \ell(\mathbf{M}, \Theta_r)$ and $\ell_v = \ell(\mathbf{M}, \Theta_v)$
  5. Interfered samples ($\ell_v > \ell_r$)
  6. Train Θ_r using $\mathbf{B}$ and $\mathbf{M}_i$
  7. Ignore Θ_v
- Θ_v has a performance hit on LSD

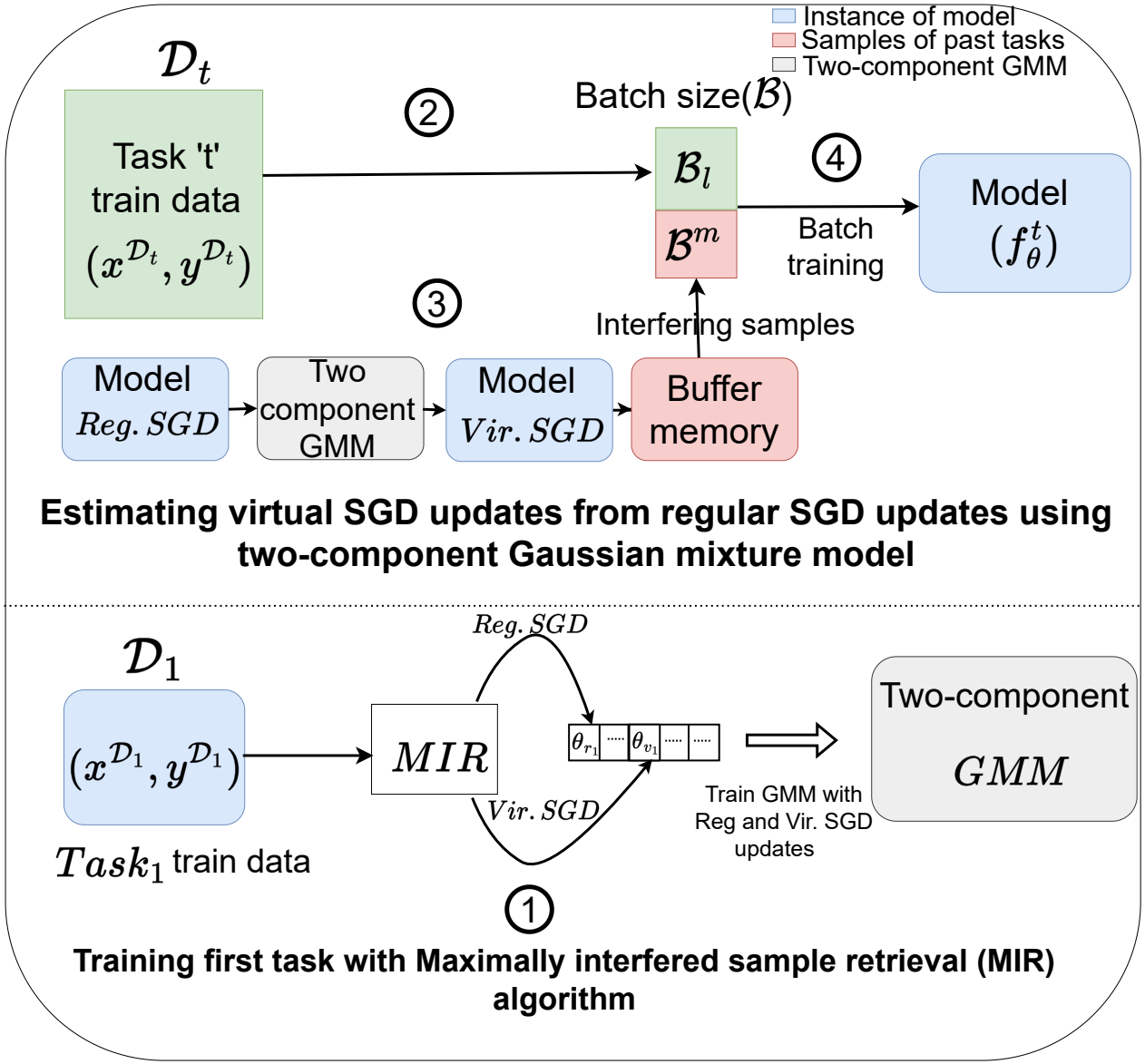


Figure 12: End-to-End training of PAPA algorithm

Table 3: Performance comparison of the proposed PAPA method with other baselines on intrusion detection and computer vision benchmarks. We report the arithmetic mean of each evaluation metric with each experiment repeated five times independently.

	MIR			PAPA			Training time (in sec.)		
Datasets	PR-AUC (A)	PR-AUC(B)	ROC-AUC	PR-AUC (A)	PR-AUC(B)	ROC-AUC	MIR	PAPA	Scalable efficiency
CTU-13	1.000	1.000	0.999	0.999	0.999	0.999	375.5	330.8	11.9%
KDDCUP'99	1.000	0.929	0.999	1.000	0.920	0.999	457.0	395.0	13.5%
NSL-KDD	0.964	0.971	0.970	0.961	0.968	0.969	25.4	21.4	15.7%
CIFAR-10	0.949	0.936	0.944	0.948	0.936	0.944	133.0	108.0	18.7%
CLEAR-100	0.839	0.793	0.817	0.845	0.793	0.823	1706.0	1209.0	29.1%
SVHN	0.979	0.972	0.976	0.981	0.974	0.978	296.0	208.0	29.7%
UNSW-NB15	0.999	0.999	0.999	0.999	0.999	0.999	499.4	350.6	29.7 %
ANOSHIFT	0.944	0.926	0.934	0.947	0.927	0.934	1300.0	900.6	30.7%
CIFAR-100	0.663	0.659	0.663	0.673	0.647	0.672	115.2	76.7	33.4%
CLEAR-10	0.953	0.933	0.942	0.943	0.927	0.932	262.9	175.4	33.2%
CICIDS-2018	0.994	0.994	0.992	0.998	0.999	0.998	9040.0	5948.0	34.2%
CICIDS-2017	0.999	0.999	0.999	0.999	0.999	0.999	316.0	188.8	40.2%

Table 4: Performance comparison of the number of regular and virtual SGD operations required for the MIR and proposed PAPA approach on benchmark datasets. Each experiment is repeated five times independently.

	MIR		PAPA		savings	
Datasets	Vir.SGD ops	Reg. SGD ops	Vir. SGD ops	Reg. SGD ops	Vir. SGD ops	Total SGD ops
NSL-KDD	1140	1578	210	1740	81.5%	28.2%
CICIDS-2017	9160	15784	550	10881	89.5%	54.17%
UNSW-NB15	11915	14638	600	12587	94.9%	50.3%
CTU-13	13235	18007	120	24658	99.0%	20.6%
KDDCUP'99	19555	17525	480	24450	97.5%	32.7%
ANOSHIFT	48825	53187	1200	58421	97.5%	41.5%
CICIDS-2018	30590	41234	1200	36605	96.1%	47.3%
SVHN	3850	7143	360	6267	90.6%	39.7%
CIFAR-10	1935	2526	180	3028	90.6%	28.0%
CIFAR-100	1700	2436	135	2253	92.0%	42.2%
CLEAR-10	500	602	40	607	92.0%	41.2%
CLEAR-100	3250	3250	320	3848	90.1%	35.8%

Formulating two dissimilar tasks learning

- Subsequent learning from MNIST, CIFAR-10 datasets

Table 5: Performance results of the MNIST+CIFAR-10 experiments

Algorithm	PR-AUC (A)	PR-AUC (B)	ROC-AUC
MIR	0.675±0.053	0.700±0.030	0.657±0.040
PAPA	0.645±0.047	0.661±0.099	0.628±0.080

How they are dissimilar ?

Table 6: Characteristics of the MNIST and CIFAR-10 datasets

Dataset	Size	No of channels	Image type
MNIST	28 X 28	1	Grayscale
CIFAR-10	32 X 32	3	RGB

Do CV tasks have any correlation to NIDS tasks ?

- Indeed, the intution holds (how ?)
- Most NIDS datasets are curated using two **sub-network** architecture
  1. Victim network
  2. Attack network

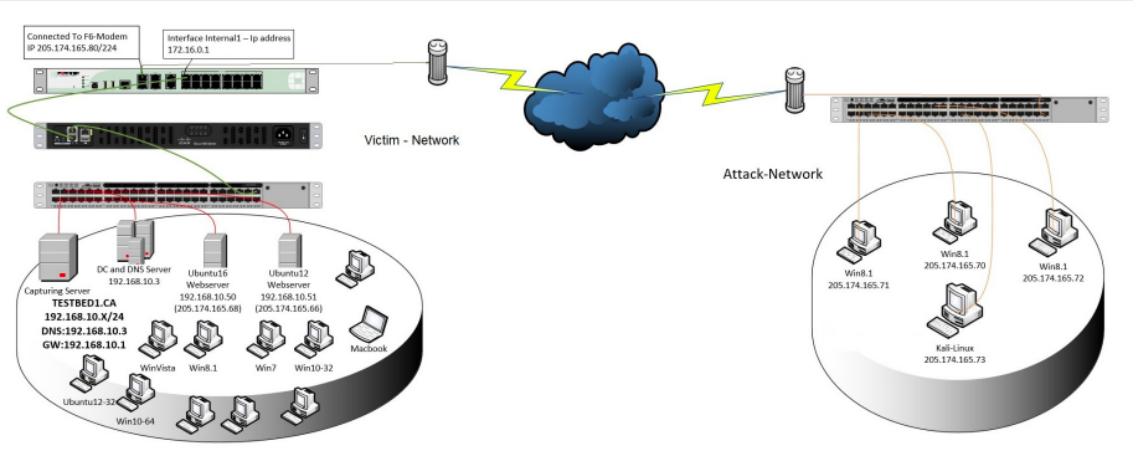


Figure 13: Testbed architecture

- ANOSHIFT curated from 5 subnets, 348 honeypots, across Kyoto university.
- Multi-diverse, spread over longer time-span

[5] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization.". In: *ICISSp* 1 (2018), pp. 108–116.

ECBRS

- Memory size
- Batch size
- Amount of benign sample to maintain

PAPA

- Batch size
- Memory size
- Impact of different first tasks on performance
- Different task order permutations

Anomaly detection datasets

- Both ECBRS and PAPA
- SMAP- Soil Moisture Active Passive
- MSL-Mars Science Laboratory Rover
- SMD-Server Machine Datase

Conclusions

- Two novel contributions aimed at improving
  1. Performance of NIDS under **severe** class imbalance
     - Extended class balanced reservoir sampling (CBRS), dubbed ECBRS
  2. Scalability- reduce the **total train time**
     - Perturbation assistance for parameter approximation (PAPA)
- ECBRS handles **severe class imbalance** better compared to CBRS
- PAPA achieves 12 to 40% **training time** saving compared to the MIR algorithm

Future directions

- Open world NIDS with *explainability*
- Semi-supervised/Unsupervised CL methods for NIDS
- Multi-class classification!!

Suresh Kumar Amalapuram
cs19resch11001@iith.ac.in
Networked Wireless Systems (NeWS lab),
Dept. of Computer Science and Engineering,
Indian Institute of Technology Hyderabad, India

Q & A

https://github.com/amalapuram/CLbasedNIDS


Chrysakis, Aristotelis and Marie-Francine Moens. "Online continual learning from imbalanced data". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1952–1961.


Dragoi, Marius et al. "AnoShift: A distribution shift benchmark for unsupervised anomaly detection". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 32854–32867.


Lisa, Liu et al. "Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018". In: *2022 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2022, pp. 254–262.


Rahaf, Aljundi and Caccia Lucas. "Online Continual Learning with Maximally Interfered Retrieval". In: *NIPS*. 2019.


Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization.". In: *ICISSp* 1 (2018), pp. 108–116.