Unsupervised Shallow Methods

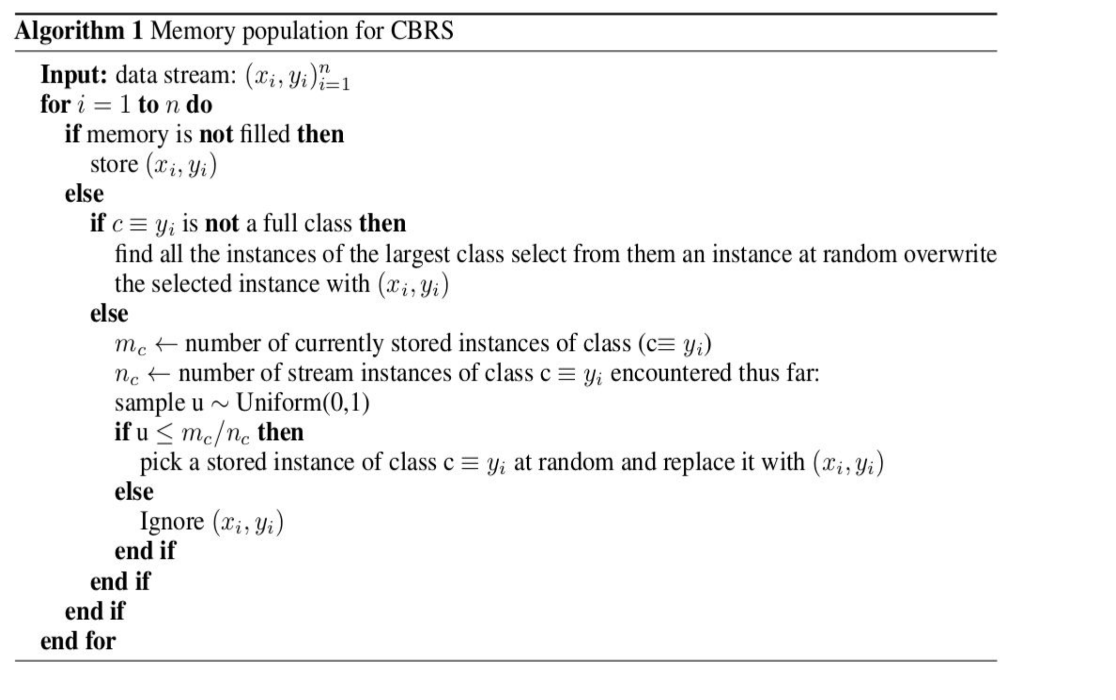
Introduction

- Network intrusion detection (NID) is a form of anomalous activity detection in communication network traffic.
- Continual learning (CL) approaches to intrusion detection aim to accumulate old knowledge while adapting to the latest threat knowledge.
- However, class imbalance and scalability issues pose significant challenges to CL-based network intrusion detection.
- The paper presents two novel approaches, Extended Class Balancing Reservoir Sampling (ECBRS) and parameter approximation (PAPA) to handle these issues.

CBRS and ECBRS

1. CBRS is a memory replay-based approach proposed in the paper to address the problem of class imbalance in continual learning for network intrusion detection.

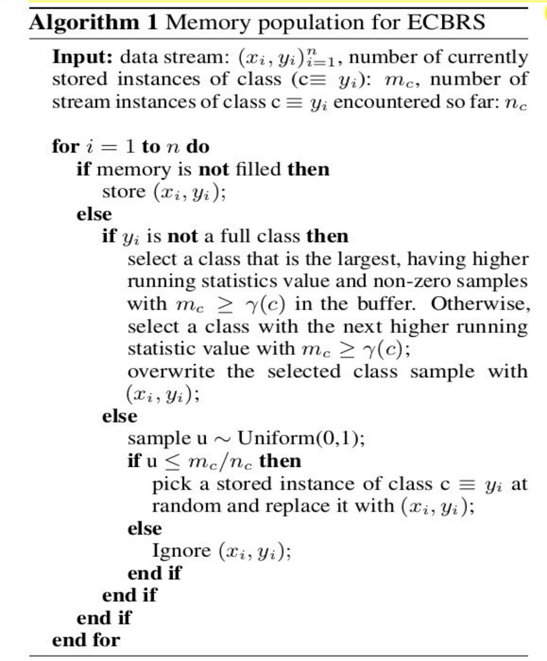
2. It is designed to undermine the majority class by maintaining global information about class imbalance, thereby ensuring that the minority class samples are replayed more frequently during training



Algorithm 1 Memory population for CBRS

```
Input: data stream: $(x_i, y_i)_{i=1}^n$
for $i = 1$ to $n$ do
  if memory is not filled then
    store $(x_i, y_i)$
  else
    if $c \equiv y_i$ is not a full class then
      find all the instances of the largest class select from them an instance at random overwrite the selected instance with $(x_i, y_i)$
    else
      $m_c \leftarrow$ number of currently stored instances of class (c$\equiv y_i$)
      $n_c \leftarrow$ number of stream instances of class c $\equiv y_i$ encountered thus far:
      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
```

CBRS and ECBRS



Algorithm 1 Memory population for ECBRS

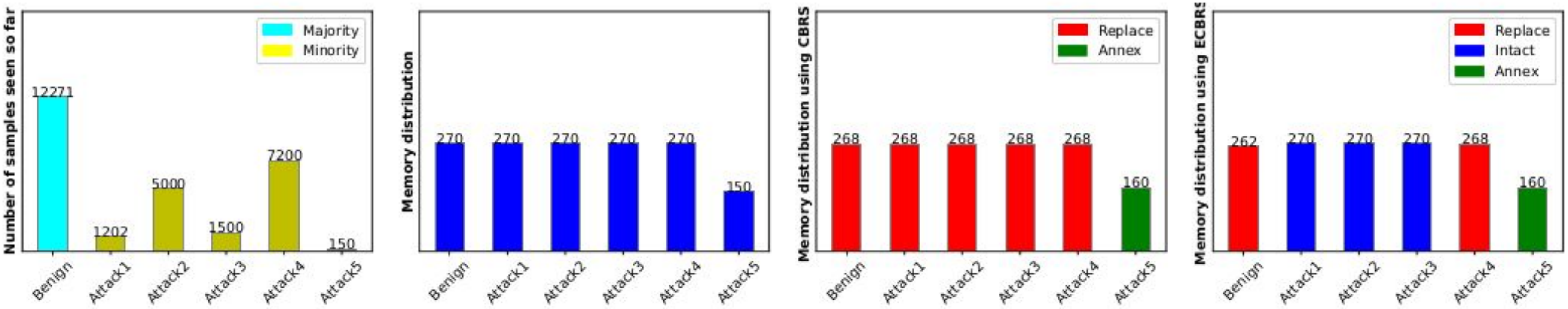
```
**Input:** data stream: $(x_i, y_i)_{i=1}^n$, number of currently stored instances of class (c$\equiv$ $y_i$): $m_c$, number of stream instances of class c $\equiv$ $y_i$ encountered so far: $n_c$

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 a class that is the largest, having higher running statistics value and non-zero samples with $m_c \geq \gamma(c)$ in the buffer. Otherwise, select a class with the next higher running statistic value with $m_c \geq \gamma(c)$;
            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
```

$$\gamma(i) = m \times w(i),$$

$$w(i) = \frac{\exp(-n_i)}{\sum_j \exp(-n_j)}$$

CBRS and ECBRS



(a) Running statistics

(b) Memory distribution

(c) CBRS

(d) ECBRS

MIR and virtual SGD updates

1. MIR (Maximally Interfered Retrieval) is a family of memory replay techniques used in continual learning, specifically for identifying and prioritizing the most interfered samples in the buffer memory.

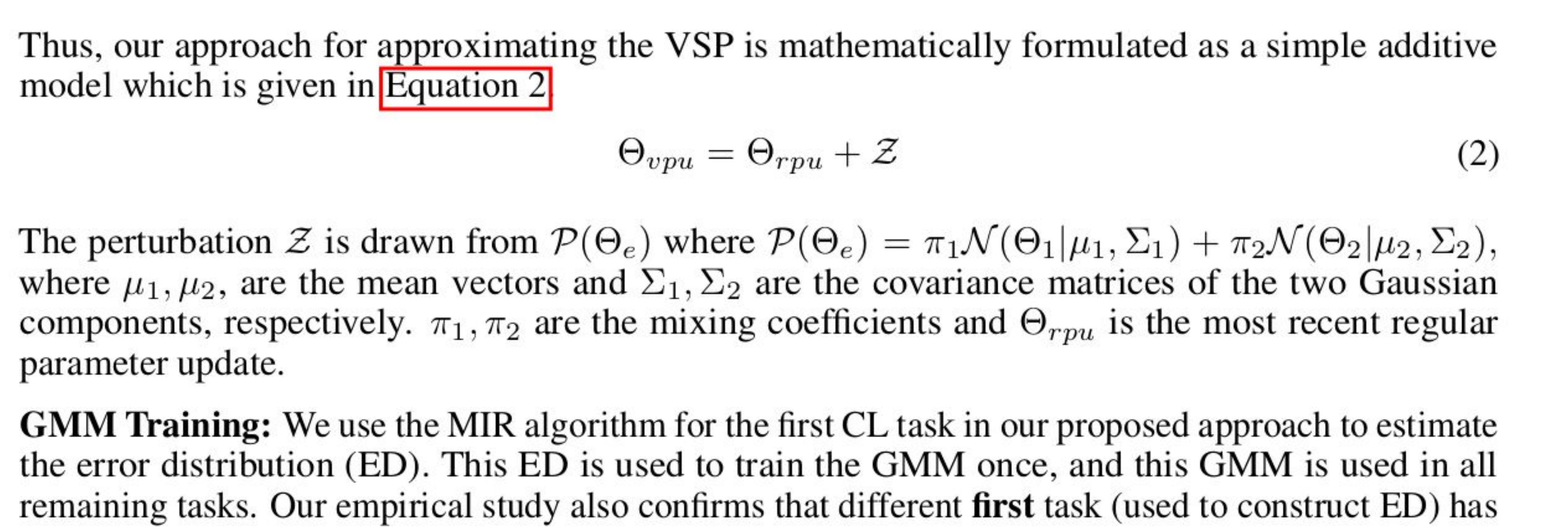
2. It quantifies the significance of samples in the buffer memory using temporary (virtual) stochastic gradient descent (SGD) parameter updates.

3. MIR is designed to address the problem of catastrophic forgetting by identifying and replaying the most interfered samples to mitigate the degradation of past tasks' knowledge over time.

PAPA

1. Perturbation Assistance for Parameter Approximation (PAPA) method is a novel approach proposed in the paper to reduce the number of VSGD update computations.
2. It is based on the Gaussian mixture model and is designed to significantly reduce the training time for methods like MIR and GMED, leading to improved scalability.
3. Experimental results demonstrate that PAPA results in training time savings of 12 to 40% compared to the baselines, making it highly suitable for large-scale training

PAPA



Thus, our approach for approximating the VSP is mathematically formulated as a simple additive model which is given in Equation 2.

$$\Theta_{vpu} = \Theta_{rpu} + \mathcal{Z} \tag{2}$$

The perturbation $\mathcal{Z}$ is drawn from $\mathcal{P}(\Theta_e)$ where $\mathcal{P}(\Theta_e) = \pi_1 \mathcal{N}(\Theta_1 | \mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\Theta_2 | \mu_2, \Sigma_2)$, where μ_1, μ_2, are the mean vectors and Σ_1, Σ_2 are the covariance matrices of the two Gaussian components, respectively. π_1, π_2 are the mixing coefficients and Θ_{rpu} is the most recent regular parameter update.

GMM Training: We use the MIR algorithm for the first CL task in our proposed approach to estimate the error distribution (ED). This ED is used to train the GMM once, and this GMM is used in all remaining tasks. Our empirical study also confirms that different **first** task (used to construct ED) has

Future Possible Extensions

- Extend to semi supervised and unsupervised binary classification.
- Exploring semi supervised techniques for intrusion detection to understand the challenges in the context of class imbalance and distribution shifts in conjunction with open-world learning, explainability, and adversarial robustness settings.