Technical Report and Additional Materials for *SHE: A Generic Framework for Data Stream Mining over Sliding Windows*

Anonymous Author(s)

November 17, 2021

Here, we briefly introduce the content of this technical report. In **Appendix A**, we explain the constraints for hardware implementation (including the reason why SWAMP cannot be deployed on hardware platforms). In **Appendix B**, we conduct a series of rigorous mathematical analysis of SHE. In **Appendix C**, we added the detailed parameter settings of several comparison algorithms mentioned at the end of Sec. 6.1 of the original paper. In **Appendix D**, we show the experiments on the impact of the parameters of SHE. The serial number of the cited literature is consistent with that of the original paper, such as **SWAMP** [8]. Important notations used in the original paper and technical report are demonstrated in Table 1.

Table 1: Notations frequently used in the original paper & technical report.

Notation	**Meaning**
t_{cur}	current time
N	size of a sliding window
M	number of cells in a data structure
T_{cycle}	time of a cleaning cycle
α	a constant parameter, $\alpha = \frac{T_{cycle}-N}{N}$
G	number of groups
w	number of cells in a group, $w = \frac{M}{G}$
d_{gid}	the time offset of the gid-th group, $d_{gid} = -\lfloor \frac{T_{cycle} \cdot gid}{G} \rfloor$
$m[gid]$	the time mark of the gid-th group

A Constraints for Hardware Implementations

Although the circuits on some advanced programmable ASICs can be designed as complex as the microprocessor, the pipeline architecture is usually preferred in order to achieve high processing speed with limited hardware resources. Therefore, when processing data streams, a hardware-friendly algorithm is supposed to be implemented as a series of pipeline stages with the following constraints:

(1) **Limited size of SRAM memory.** Memory access is usually the limitation of the processing speed on most hardware platforms. SRAM provides faster memory access but it is more expensive and therefore the memory size of SRAM is typically small.

(2) **Single stage memory access.** A read-write hazard may be caused when two stages access the same memory region simultaneously. Therefore, each memory region is supposed to be accessed once when an item is going through the pipeline.

(3) **Limited concurrent memory access.** Each stage can access one memory address with limited size. In the other word, it is not suitable to access the entire memory block or a large amount of memory in a single stage.

The constraints mentioned above are not met when using **SWAMP** [8] to process data streams. First, the Tiny Table used by **SWAMP** records the fingerprints of all the arriving items, and thus the memory usage is not affordable

when dealing with a large sliding window. Meanwhile, buckets, of which the Tiny Table consists, are somehow tied together. When an item is inserted into a filled bucket, the bucket will expand to its adjacent buckets, and the domino effect may occur, which leads to an unlimited concurrent memory access. Even if the domino effect is constrained, the operation of the Tiny Table is still too complex to be finished within one memory access. There are three fields in a bucket, each of which is probably modified during one single insertion. However, an extra deletion occurs when the space of the bucket is fully used and it is not allowed to expand to other buckets. Therefore, the changes of the three distinct fields may affect each other and therefore they can not be modified sequentially.

B Mathematical Analysis

Here, we analysis (1) the error caused by the on-demand cleaning (Sec. 3.2, page 3) and (2) the error bounds in different tasks. For time-based sliding window, we assume that the items arrive at a uniform speed. So we only need to analyse SHE based on counter-based sliding window.

B.1 Error Bound of On-demand Cleaning

In this part, we analyze the accuracy of on-demand cleaning. If every group can be updated in one time cycle T_{cycle}, the on-demand update of the group is zero error. We compute the probability that a group fails to be updated, *i.e.*, fails to be mapped by any item in a sliding window. Let n be the number of groups that fail to be updated and let $\overline{C}$ be the cardinality of a sliding window. We assume that the cardinality of the stream in a cleaning cycle, whose size is $T_{cycle} = (1+\alpha)N$, is $(1+\alpha)\overline{C}$. Thus, the number of cells that are updated in the cleaning cycle is $(1+\alpha)\overline{C}H$, where H is the number of inserted cell(s) in each insertion. The expectation of the number of groups that fail to be updated in one sliding window is $E = G \cdot (1-\frac{1}{G})^{(1+\alpha)\overline{C}H} = G \cdot e^{-\frac{(1+\alpha)\overline{C}H}{G}}$. To make $E \leqslant \varepsilon$ where ε is a small constant, we get the following inequality :

$$G \ln G * \frac{1}{(1+\alpha)\overline{C}H} \leqslant \varepsilon \tag{1}$$

In practice, when we determine an ε and compute a G, we can get the size of each group $w = \frac{M}{G}$.

B.2 False Positive Rate of the Membership Task

In this part, we estimate the false positive rate of the Bloom filter [12] using our SHE (SHE-BF), and provide a equation to determine the value of α. Similar to the Bloom filter, the SHE-BF has a one-side error (*i.e.*, only false positive error but no false negative error). Let $R = \alpha + 1$. The estimated cardinality of data streams in a cleaning cycle of size $T_{cycle} = rN$ ($r \in [0, R)$) is rC. For a group whose age is rT, the expectation of the proportion of 0 bits in the group is $P_0(r) = (1-\frac{1}{w})^{CHr/G}$. For fixed w,G,C,H, let $Q = (1-\frac{1}{w})^{CH/G}$. Then we get the false positive rate:

$$FPR_{(R)} = \left[1 - \frac{\int_1^R P_0(r)\mathrm{d}r}{R}\right]^H = \left[1 - \frac{(Q^R - Q)}{\ln(Q)R}\right]^H$$

Let $g(R) = \frac{(Q^R - Q)}{R}$. As H and Q are fixed and $\frac{1}{\ln Q} \leqslant 0$, we can minimize $FPR_{(R)}$ by minimizing $g(R)$. To minimize $g(R)$, we take the derivative of $g(R)$ with respect to R:

$$\frac{\mathrm{d}g}{\mathrm{d}R} = Q^R \cdot [R\ln(Q) - 1] + Q$$

From the above equation, we can see that $\frac{\mathrm{d}g}{\mathrm{d}R}$ is monotonically increasing when $R \in [0, +\infty)$. Next, we solve the equation $\frac{\mathrm{d}g}{\mathrm{d}R} = 0$, and let R_0 denote the solution of the equation. Then, the optimal α is:

$$\alpha = R_0 - 1 \tag{2}$$

B.3 Error Bounds of cardinality estimation

In this part, we analyze the accuracy of SHE-BM (Bitmap [26] using SHE), SHE-HLL (HyperLogLog [19] using SHE), and SHE-MH (MinHash [13] using SHE), and offer two error bounds with respect to α for the three algorithms, respectively.

SHE-BM: For SHE-BM, we analyze its error bound. Then, we discuss the value of α for stable performance. Let $F(x)$ denotes the cardinality of data streams in the sliding window of size x (*i.e.*, the recent x items). Our goal is to estimate the cardinality when $x = T$, *i.e.*, $C = F(T)$. Suppose there are m_ℓ bits whose ages are legal in the SHE-BM, and u bits of value 0 among the m_ℓ bits.

We first analyze the over-estimation case. When the age of a group, *i.e.*, x, is within $[(1-\alpha)T, T)$, the largest possible valve of $F(x)$ is C. When the age of a group is within $[T, (1+\alpha)T)$, the largest estimation of the cardinality is $C + (x - T)$. Therefore, we get the upper bound:

$$F_{upper}(x) = \begin{cases} C, & \textbf{if } (1-\alpha)T \leqslant x < T \\ C + (x - T), & \textbf{if } T \leqslant x < (1+\alpha)T \end{cases}$$

Then, we can get a lower bound of u, *i.e.*, the number of 0 bits among the m_ℓ legal bits.

$$\begin{aligned} E_{lower}\left[\frac{\widehat{u}}{m_\ell}\right] &= \frac{1}{2}\left[\frac{\int_0^{\alpha T}(1-\frac{1}{m_\ell})^{C+x}\mathrm{d}x}{\alpha T} + \left(1-\frac{1}{m_\ell}\right)^C\right] \\ &\geqslant e^{-\frac{C}{m_\ell}} \cdot \left(1 - \frac{\alpha T}{4m_\ell}\right) \end{aligned}$$

Therefore, the upper bound of the estimated cardinality $\widehat{C}$ is:

$$E[\widehat{C}] = -m_\ell \cdot \ln\left(E_{lower}\left[\frac{\widehat{u}}{m_\ell}\right]\right) \leqslant C + \frac{\alpha T}{4}$$

In the same way, we can get the lower bound of the estimated cardinality $\widehat{C}$:

$$E[\widehat{C}] \geqslant C - \frac{\alpha T}{4}$$

Finally, the error bound of SHE-BM is:

$$\left|\frac{E[\widehat{C}] - C}{C}\right| \leqslant \varepsilon = \frac{\alpha T}{4C} \tag{3}$$

Therefore, we can adjust the error bound ε by setting α.

From Equations (3), we can see that the error bound of $E[\widehat{C}]$ is tighter when α is smaller. However, α cannot be too small, because a small α can lead to a large variance for $E[\frac{\widehat{u}}{m_\ell}]$, and thus lead to a large variance for $E[\widehat{C}]$. Specifically, suppose p is the true proportion of 0 bits in legal bits (*i.e.*, the bits in legal groups), then the variance of $E[\frac{\widehat{u}}{m_\ell}]$ is:

$$Var\left(E\left[\frac{\widehat{u}}{m_\ell}\right]\right) = \frac{p}{m_\ell}$$

Therefore, $m_\ell = \frac{2\alpha}{1+\alpha}m = (2 - \frac{2}{1+\alpha})m$ cannot be too small, and thus α cannot be too small.

SHE-HLL: The analyzing procedure is similar to that of SHE-BM. For a group, let ρ_{max} be the expectation of the position in which the leftmost "1" is. Then we can get an upper bound of $\widehat{\rho}_{max}$:

$$E[\widehat{\rho}_{max}] \leqslant \frac{1}{2} \cdot \left[log_2\left(\frac{C}{G} + \frac{\alpha T}{2G}\right) + \log_2\left(\frac{C}{G}\right)\right]$$

Then, the upper bound of the estimated cardinality $\widehat{C}$ is:

$$E[\widehat{C}] \leqslant C + \frac{\alpha T}{4}$$

In the same way, we can get the lower bound, and finally we get the following error bound:

$$\left|\frac{E[\widehat{C}] - C}{C}\right| \leqslant \varepsilon = \frac{\alpha T}{4C} \cdot \left[1 + O\left(\frac{\alpha T}{C}\right)\right] \tag{4}$$

SHE-MH: Let $F(x)$ be the similarity between two streams in the sliding window of size x, and $S = F(T) = \frac{S_\cap}{S_\cup}$ be the similarity of the two streams when the sliding window size is T, where $S_\cap$ and $S_\cup$ are the size of the intersection set and the union set of the items in the two streams, respectively. Then, we compute the error bound of $E[\widehat{S}]$. The worst case of over-estimation is that $F(x)$ sharply decrease when $x \in [(1-\alpha)T, T]$ and sharply increased when $x \in [T, (1+\alpha)T]$. In this situation, we overestimate the similarity and therefore the upper bound of the estimated similarity is:

$$\begin{aligned} E[\widehat{S}] =& \frac{1}{2}\left(\frac{\int_0^{\alpha T}\frac{S_\cap}{S_\cup - 2x}\mathrm{d}x}{\alpha T} + \frac{\int_0^{\alpha T}\frac{S_\cap + x}{S_\cup + x}\mathrm{d}x}{\alpha T}\right) \\ \leqslant& \frac{S_\cap}{S_\cup} + \left[\frac{1}{4}\varepsilon + \frac{1}{6}\varepsilon^2 + O(\varepsilon^3)\right], \quad \text{where } \varepsilon = \frac{2\alpha T}{S_\cup}. \end{aligned}$$

We can get the lower bound of $E[\widehat{S}]$ in the same way, and the final error bound of $E[\widehat{S}]$ is:

$$\left|E[\widehat{S}] - S\right| \leqslant \frac{1}{4}\varepsilon + \frac{1}{6}\varepsilon^2 \tag{5}$$

According to Equation (5), the bias $|E[\widehat{S}] - S|$ can be bounded by ε, where ε is a small constant related to α. Therefore, we can have a tight error bound by adjusting α.

C Supplement to the Experimental Setup

For TSV [20], we use the 64-bit timestamp. For CVS [25], we set the maximum value of its counter to 10. For SWAMP [8], we use its DISTINCTMLE estimator.

SHE-HLL: We compare SHE-HLLwith SHLL [14]. For the SHLL, we store the 64-bit timestamp. For both of algorithms, we calculate 32-bit hash values and store the numbers of leading 0 of these hash values in 5-bit cells. We set the window size to 2^{21} because HyperLogLog [19] is usually used to estimate massive cardinality.

SHE-CM: We compare SHE-CM with two algorithm: ECM [24] and SWAMP. For ECM, we set the number of hash function to 4. For SHE-CM, we set the number of the hash function to 8.

SHE-BF: We compare SHE-BF with three algorithms: TBF [27], TOBF [21], and SWAMP. For TBF, we set the size of each counter to 18 bits and the number of hash functions to 8. For TOBF, we use the 64-bit timestamp. For SWAMP, we use its ISMEMBER estimator. For SHE-BF, we fix the number of hash functions to 8. α is determined according to Equation (2), which is roughly 3.

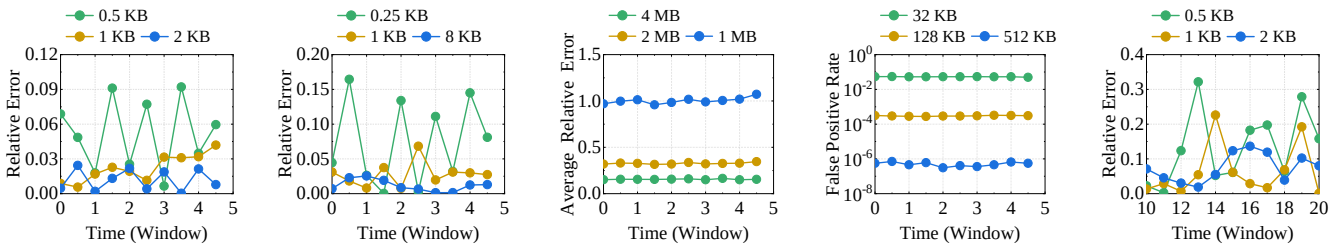
SHE-MH: We compare two algorithms: the straw-man MinHash and SHE-MH. The straw-man MinHash is the modified MinHash [13] by adding a 64-bit timestamp for each pair of counters to indicate if the counters need to be cleaned. The outputs of hash functions used in both algorithms are 24-bit integers.

D Impact of Parameters

D.1 Common Parameters

Performance *vs.* time (Fig. 1): We find that the SHE is stable as time goes by and the window slides. We test our algorithms every half window on the same stream and we test the algorithms using three different sizes of memory. When given enough memory, the performance is stable especially for SHE-BF and SHE-CM.

Performance *vs.* window size (Fig. 2): We find that the SHE is stable to the window size when other parameters are fixed. We test our algorithms for three different sizes of memory. The performance of the SHE is actually similar to the ideal goal. For example, SHE-HLL and SHE-MH keep almost constant ARE because the original algorithms are not sensitive to the size of data.



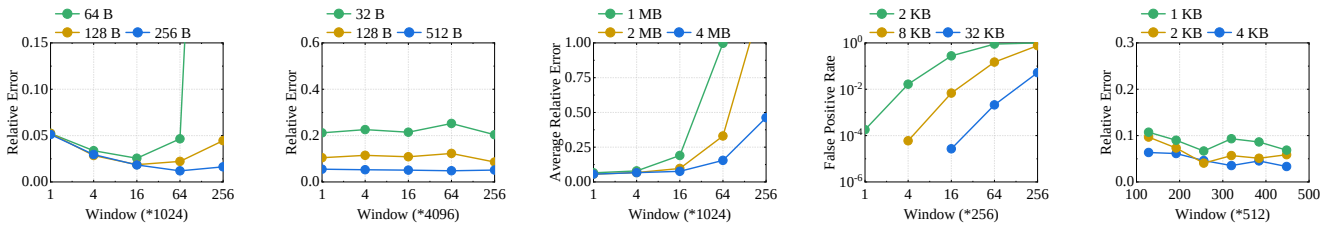
(a) Cardinality (Bitmap) (b) Cardinality (HLL) (c) Frequency (d) Membership (e) Similarity

Figure 1: The stability of SHE as the window slides with time.

Performance *vs.* α (Fig. 3): The experimental results of the other three algorithms (SHE-CM, SHE-HLL, and SHE-MH) are similar to SHE-BM. Therefore, we only show the performance of the SHE-BF and SHE-BM. The optimal α is computed according to Equation (2). As shown in Fig. 3(a), the SHE-BF using the optimal α performs well on the real dataset. For SHE-BM and other SHE-algorithms, the SHE performs well when α is from 0.2 to 0.4 as an empirical setting.

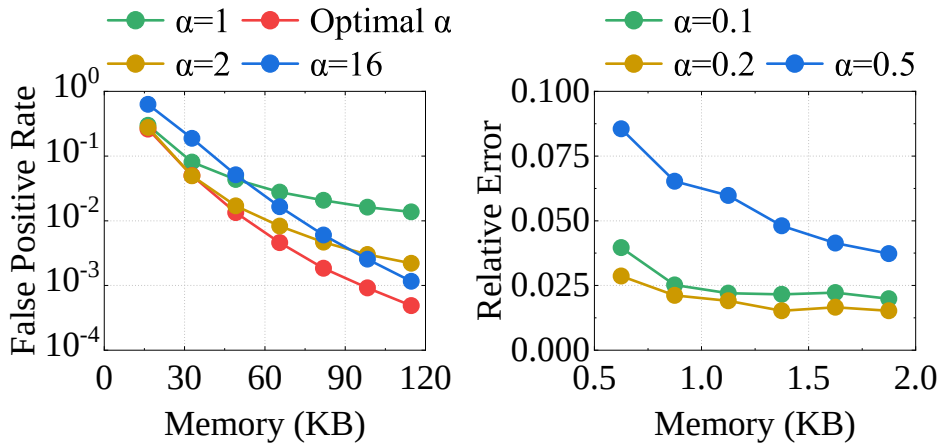
D.2 Parameters of SHE-BF

FPR *vs.* time age (Fig. 4(a)): The figure shows that the FPR becomes stable when the item age is increasing. The item age is the time spans from the item's arrival to the current time. We test the SHE-BF on Distinct Stream and repeated our experiment for $10,000$ times. As the item age increases, the FPR decreases at a nearly exponential speed until the item age is larger than the size of relaxed window.



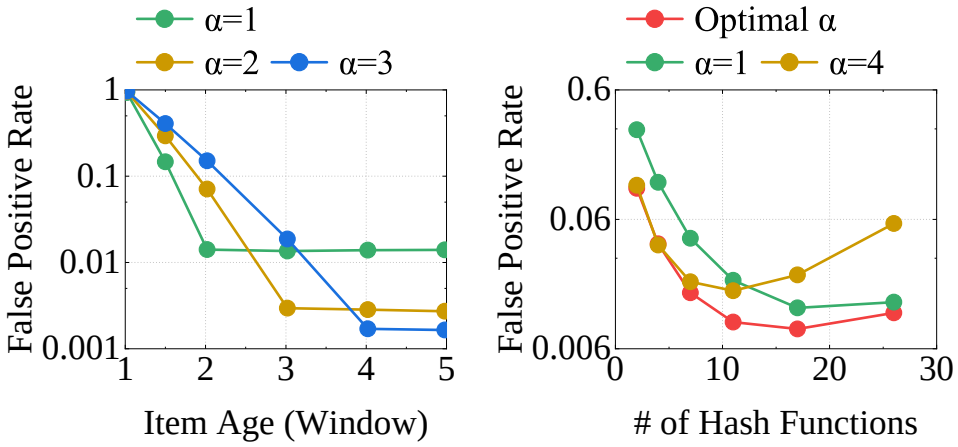
(a) Cardinality (Bitmap) (b) Cardinality (HLL) (c) Frequency (d) Membership (e) Similarity

Figure 2: The adaptation for different window size. When the 'Window(*k)' is x, it means that the window size is x*k.



(a) SHE-BF (b) SHE-BM

Figure 3: Performance *vs.* α.



(a) FPR *vs.* Item Age. (b) FPR *vs.* # of Hash Functions.

Figure 4: Parameters of SHE-BF.

FPR *vs.* number of hash functions (Fig. 4(b)): The optimal α depends on the number of hash functions according to Equation 2. We find that the optimal α for different number of hash functions estimated by Equation 2 works well on Distinct Stream, which is the worst case for Bloom filter as mentioned in Sec. 6.1.