Appendix – Timely Detection and Mitigation of Stealthy DDoS Attacks via IoT Networks

Keval Doshi, Yasin Yilmaz, and Suleyman Uludag

Proof of Theorem 1

Consider a hypersphere $\mathcal{S}_t \in \mathbb{R}^d$ centered at $\boldsymbol{x}_t^n$ with radius L_t^n, the kNN distance of $\boldsymbol{x}_t^n$ with respect to the training set $\mathcal{X}_{M_2}^n$. The maximum likelihood estimate for the probability of a point being inside $\mathcal{S}_t$ under f_0 is given by k/M_2. It is known that, as the total number of points grow, this binomial probability estimate converges to the true probability mass in $\mathcal{S}_t$ in the mean square sense [1], i.e.,

$$k/M_2 \xrightarrow{L^2} \int_{\mathcal{S}_t} f_0(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

as $M_2 \to \infty$. Hence, the probability density estimate

$$\hat{f}_0(\boldsymbol{x}_t^n) = \frac{k/M_2}{V_d (L_t^n)^d},$$

where $V_d (L_t^n)^d$ is the volume of $\mathcal{S}_t$, converges to the actual probability density function, $\hat{f}_0(x_t^n) \xrightarrow{p} f_0(x_t^n)$ as $M_2 \to \infty$, since $\mathcal{S}_t$ shrinks and $L_t^n \to 0$. Similarly, considering a hypersphere $\mathcal{S}_{(\alpha)} \in \mathbb{R}^d$ around $\tilde{\boldsymbol{x}}_{(\alpha)}^n$ which includes k points within its radius $\tilde{L}_{(\alpha)}^n$, we see that as $M_2 \to \infty$, $\tilde{L}_{(\alpha)}^n \to 0$ and

$$\hat{f}_0(\tilde{\boldsymbol{x}}_{(\alpha)}^n) = \frac{k/M_2}{V_d (\tilde{L}_{(\alpha)}^n)^d} \xrightarrow{p} f_0(\tilde{\boldsymbol{x}}_{(\alpha)}^n).$$

Assuming a uniform distribution

$$f_1(\boldsymbol{x}) = f_0(\tilde{\boldsymbol{x}}_{(\alpha)}^n), \ \forall \boldsymbol{x},$$

we conclude with

$$\log \frac{\frac{k/M_2}{V_d (\tilde{L}_{(\alpha)}^n)^d}}{\frac{k/M_2}{V_d (L_t^n)^d}} = d \left[\log L_t^n - \log \tilde{L}_{(\alpha)}^n \right] \xrightarrow{p} \log \frac{f_1(\boldsymbol{x}_t^n)}{f_0(\boldsymbol{x}_t^n)}$$

as $M_2 \to \infty$.

Proof of Theorem 2

In online testing (see lines 6-11), the most expensive part is to compute D_t^n, in particular L_t^n. And within L_t^n the expensive part is to find the kth nearest neighbor, which is $O(M_2 d)$ if computed straightforwardly by computing the distance of test point to all M_2 training points. The space complexity of the algorithm is due to storing M_2 training points, each of which is d-dimensional, i.e., $O(M_2 d)$. Note that the both time and space complexity of the mitigation part shown in lines 13-23 is $O((T - \tau + 1)d)$ where $T - \tau + 1$ is a bounded number close to the detection delay, typically much smaller than M_2. In training, to compute $\tilde{L}_{(\alpha)}^n$ shown in line 4, kth nearest neighbor among M_2 points are computed for each of M_1 points, requiring $O(M_1 M_2 d)$ computations. However, training is performed once offline, so the complexity of online testing is usually critical for scalability.

References

[1] Alan Agresti. *An introduction to categorical data analysis.* Wiley, 2018.

This work was supported in part by the U.S. National Science Foundation under the Grant CNS-1737598 and in part by the SCEEE-17-03 Grant.

K. Doshi and Y.Yilmaz are with the Electrical Engineering Department, University of South Florida, Tampa, FL USA (e-mail: keval-doshi@mail.usf.edu, yasiny@usf.edu).

S. Uludag, is with the Department of Computer Science, University of Michigan - Flint, MI USA (e-mail: uludag@umich.edu).