Incremental Federated Host Embeddings for Network Telescopes Traffic Analysis

Incremental Federated Host Embeddings for Network Telescopes Traffic Analysis

Kai Huang, Luca Gioacchini, Marco Mellia, Luca Vassio
Politecnico di Torino
Turin, Italy
first.last@polito.it

***Abstract*—Network telescopes are ranges of IP addresses with nothing connected. They are contacted by botnets and scanners that look for possible victims. Each telescope exposes a partial view, and merging the information with that coming from other telescopes is fundamental. Machine learning allows us to build models to solve classification tasks automatically. However, the continuous evolution of traffic calls for a continuous update of such a model. This work explores applying collaborative Artificial Intelligence solutions via Federated Learning (FL) to build a global model without sharing the raw (and sensitive) data, also limiting data exchange. We leverage a two-stage pipeline: (i) a self-supervised upstream task generates and updates an incremental compact representation of the senders hitting the telescope; (ii) such embeddings serve as input for a downstream classification task to identify possible offenders. We compare the embedding that a single telescope generates with those obtained via FL from data collected by multiple telescopes and evaluate the benefits of the incremental approach. We show that FL can produce embeddings of better quality than a single network telescope can, increasing the model accuracy (+6%) and coverage (+12%) while limiting the amount of data exchanged (from GBs to MBs).**

***Index Terms*—Federated Learning, Network Telescope, Network Traffic Analysis, Host Embeddings, Privacy.**

I. Introduction

Network monitoring plays a critical role in cybersecurity as it allows the continuous observation and analysis of traffic to detect and mitigate potential threats, minimising the risk of cyber-attacks. The humongous amount and the continuously evolving nature of traffic reaching networks require the adoption of scalable solutions based on Artificial Intelligence (AI) to assist network and security analysts in unveiling hidden traffic patterns and coordinated malicious activities [1], [2].

Network telescopes, or telescopes or darknets, are valuable monitoring tools made of ranges of IP addresses not hosting any services [3]. They only receive traffic resulting from misconfigured services, routine scans or botnet activities. For this, they represent a privileged point of view for cybersecurity applications. As such, security providers deploy different network telescopes as sensors. Yet, each telescope offers a unique but partial view of internet activities [4].

To extract actionable information, a common recent trend sees the employment of an AI-based 2-stage pipeline [2], [5], [6] as shown in Figure 1: data arrives to the sensor in batches, e.g., every day or hour; given a new batch of data, a *self-supervised upstream task* updates embeddings, i.e. compressed representations of input data in a latent space with no need for ground truth. Such embeddings serve as input to specialised models to solve specific problems in the *downstream tasks*. In fact, the adoption of NLP-based embeddings successfully enables the identification of new attacks and threats from network telescopes [2], [7]–[9].

Given each network telescope provides a different view [4], security providers can benefit greatly from collaboration by sharing data and intelligence collected by different network telescopes. However, the volume of network traffic challenges the sharing of data, where even a small /24 telescope can observe millions of packets in one day [9]. Furthermore, there are serious privacy and security concerns associated with network telescope data, as CAIDA, the provider of the world's largest network telescope, points out [10]. In fact, some providers share telescope data by hashing senders' IP addresses [11] which greatly reduces the value of the data.

The recent development of Federated Learning (FL) techniques paves the road for collaborative solutions allowing the extraction of information across different networks without requiring the sharing of raw data [12]. FL allows us to learn a common model by aggregating locally-computed updates. In addition, the continuous evolution of the attack patterns calls for a continuous and incremental update of the model. From this comes the need to incrementally update the model, which FL naturally favours. In cybersecurity literature, FL solutions gained traction to limit data sharing and build global intrusion detection models [13], risk intelligence systems [14], monitor IoT networks [15], perform DDoS attack detection and classification [16]. The authors of [17] focus on creating representations for the occurrence of a handful of malware activities starting from telescope traffic. Our work differs as we aim to learn comprehensive representations from network telescope traffic that fit different downstream tasks.

We adopt the 2-stage pipeline [6] relying on i-DarkVec [2], an NLP-based methodology to represent senders targeting network telescopes. We extend i-Darkvec to work with multiple network telescopes through the FL framework [12]. We evaluate the goodness of the produced embeddings by formulating a sender classification problem. We gauge the benefit of the FL approach by varying the number and the size of involved telescopes. Finally, we test the improvement obtained by a continuous incremental update of the embedding.

Results show that the FL collaboration between multiple

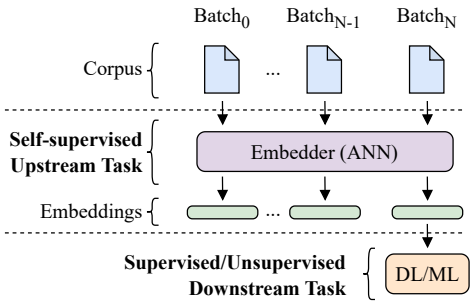


Fig. 1: Incremental two-stage pipeline for temporal analysis.

telescopes allows us to improve both the i) quality and ii) coverage of the AI pipeline: we obtain a better model which extends the visibility of each telescope thanks to the information provided by other telescopes. Fundamental to this is the ability to continuously and incrementally update such models.

II. Network Telescope Traffic

In this work, we rely on data collected from two network telescopes. The first one is a /24 network telescope situated in our university campus network, and the second one is a /19 network telescope operated in a Brazilian academic network.

We collect one month of network telescope traffic, from 2021-05-01 to 2021-05-31, observing more than 64 million packets sent by 532 thousand senders in the /24 network telescope and more than 1.5 billion packets sent by 3 million senders in the /19 one. We remove from the original collection senders sending less than 5 daily packets [2], retaining 15% and 30% of the observed senders in the /24 network telescope and the /19 one respectively.

a) Supervised Task and Ground truth: As supervised downstream task, we perform a host classification assigning senders to known classes characterised by coordinated behaviours. We leverage a ground truth representing classes of senders whose coordination is known a priori relying on two data sources: (i) the presence of fingerprints of Mirai-like malwares observed in received packets [18], and (ii) publicly available information retrieved from online repositories of acknowledged internet scanners[1] – i.e. non-hostile senders performing scanning activities. The final ground truth has 13 classes covering $\approx$ 13 thousand ($\approx$ 23 thousand) of the senders observed in the whole month within the /24 network telescope (/19 network telescope). We mark all the remaining senders *Unknown*, ending up with a highly unbalanced ground truth – e.g. thousands of senders exhibit the Mirai-like label, while only hundreds or dozens belong to classes of acknowledge scanners.

b) Sampling subnets: When investigating the potential of FL approaches in different scenarios, we design experiments relying on network telescopes of different sizes. We consider our campus telescope as the provider d_1, and mimic different network telescopes by splitting addresses of the /19 Brazilian network telescope into sub-telescopes. Namely, we extract multiple non-adjacent subnets whose size ranges from /20 to /28. We refer to one of the sampled /24 network telescopes as d_2 when 2 providers are involved.

Next, we investigate the performance of the collaborative embeddings in a downstream supervised classification task.

III. Learning Federated Host Embeddings

We generate host embeddings through state-of-the-art approach i-DarkVec [2], which relies on Word2Vec [19] and incremental training. We assume that the reader is familiar with Word2Vec. We consider the traffic collected by a network telescope which receives traffic from external hosts, or *senders* identified by their IP addresses. Our goal is to identify some common patterns and even coordinated attacks groups of senders contribute to, e.g., coordinated botnets or network scanners. Given a batch of packets, we extract the sequences of senders that send packets to the same TCP/UDP ports. Analogously to NLP, senders represent "words", and their sequence represents "sentences". We feed the generated sequences as input to Word2Vec to produce senders' embeddings. This process projects the senders in a latent space such that senders co-occurring in time when targeting similar ports appear close in the latent space.

Formally, given a *vocabulary* of senders $V = \{v_1, v_2, v_3, \ldots\}$ and the sequences of packets they send, i.e. the *corpus* C, we map each entity $v \in V \rightarrow u \in \mathbb{R}^E$ where u is the embedding of v in the E dimensional space. The function $e : V \rightarrow \mathbb{R}^E$ is the embedding function (i.e. Word2Vec) that we train by giving in input the corpus C in a self-supervised manner using the masked language technique. Finally, given the function e, let $\mathbf{X} = [e(v)]_{v \in V}$ be the matrix of embeddings for all senders in V, i.e., $\mathbf{X} \in \mathbb{R}^{|V| \times E}$.

We call the scenario in which we generate embeddings from a single network telescope *local* approach.

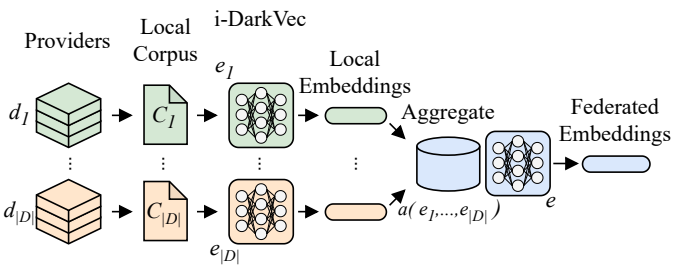
A. Incremental update of the embeddings

Given the telescope continuously collects packets over time, we propose to leverage an incremental learning approach to continuously update the embeddings: Instead of retraining from zero the Word2Vec model each time we have new data, we perform a model update by fine-tuning for just 1 epoch the model with the new batch of data. In detail, at each timestep t_{i+1}, we obtain a new batch of data from which we generate a new corpus and incrementally update the embeddings starting from the weights computed at the t_i timestep. This strategy speeds up the embeddings learning and lets the system weigh newer information automatically.

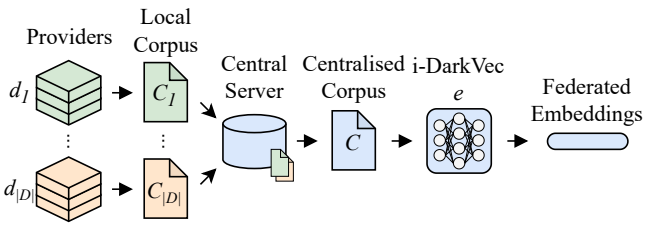
This requires updating the vocabulary and modifying the neural network topology. In fact, at each epoch, the telescope observes possibly new senders. This makes the vocabulary grow and implies the addition of a new neuron for each new sender in the input and output layers[2].

[1] https://gitlab.com/mcollins_at_isi/acknowledged_scanners

[2] As required by the Word2Vec One-Hot-Encoding of the words [19]



(a) FL approach.



(b) Centralised approach.

Fig. 2: Overview of the adopted collaborative approaches.

B. Federated approach

To extend i-DarkVec to multiple networks, we rely on the idea of *FedAvg* [12], an efficient FL algorithm allowing distributed training among a large number of clients[3]. In FedAvg, a central server distributes model parameters to the clients and aggregates their updates. We provide an overview of the FL approach in Figure 2a.

Given a set of network telescopes D (the providers), each provider $d \in D$ generates the corpus C_d according to the traffic it observes, updates its local embedding function e_d and generates the local embedding matrix $\mathbf{X}_d \in \mathbb{R}^{|V_d| \times E}$, where V_d is the local vocabulary. Then, the central server receives the local models e_d and vocabularies V_d. The server aggregates the local models into a federated model that can be applied on any sender observed in at least a vocabulary. In formulas, the server aggregates the functions $e_1, \ldots, e_{|D|}$ into a new function $e : \bigcup_d V_d \rightarrow \mathbb{R}^E$. Finally the server sends back e to each provider. Notice that now, for each provider d, after the aggregation, $e_d = e$. This update procedure repeats for multiple rounds.

We use as the aggregation function a weighted average between the local embeddings produced by each provider.[4] Notice that some local embeddings might not be defined for some providers. The weighted average is only performed among defined embeddings.

Formally, we define the final federated embedding for a host $v \in \bigcup_d V_d$ as

$$e(v) = \frac{\sum_{d: v \in V_d} w_{v,d} \cdot e_d(v)}{\sum_{d: v \in V_d} w_{v,d}}$$

where $w_{v,d} \in \mathbb{R}$ is the weight referred to the local embeddings of sender v observed in network telescope d.

Note that this entails the definition of a common vocabulary and a consequent modification of the global neural network topology. In fact, each provider observes a different and growing set of senders, and thus a different vocabulary. The server receives each client changes, computes and redistributes the global vocabulary $V_{tot} = \bigcup_d V_d$ and the new neural network topology where a new neuron is added for each new element in the vocabulary at the input and output layers.

The frequency of senders appearing in the corpus causes their embeddings to be updated at different rates. To account for this, we use a network-wise weighting scheme: Observing more senders means acquiring more information, we aggregate the local models favouring the embeddings produced by providers that have larger telescopes. Namely, we compute a unified weight to all the senders active in a telescope d as the size of the local vocabulary V_d. Thus, for each host $v \in V_d$ the weight is equal to $w_{v,d} = |V_d|$. Note that for those $v \in V_{tot},\ v \notin V(d)$ we consider $e_d(v) = \{0\}$.

C. Centralised approach

We compare the FL solution with a centralised approach baseline overviewed in Figure 2b. Here, each provider $d \in D$ builds its local corpus C_d and sends it to a central server. The server obtains the centralised corpus as the concatenation of the corpora of each provider, $C = \bigcup_{d \in D} C_d$. The server produces the final collaborative embeddings by training or updating e with the sequences of senders appearing in C.

IV. Supervised Classification Task Result

A. Methodology

Motivated by the assumption that high-quality embeddings can project senders of the same class (i.e. belonging to one of the coordinated groups of the ground truth) into adjacent regions of the latent space, we perform our downstream classification task relying on a simple k-Nearest-Neighbours (k-NN) classifier. It assigns each sender to the most frequent label through majority voting among the classes of the k nearest neighbours in the embedding space. Thus, the more compact the regions of embeddings of senders engaged in similar activities, the better the classification performance. We use *cosine distance* to measure distance among embeddings.

We account for the lack of ground truth by adopting a Leave-One-Out validation approach on the senders active in our collection. We use the Marco average F1-score to address the unbalancing among ground truth classes. Since we cannot verify the characteristics of the *Unknown* senders, we consider them in k-NN computation only but do not report classification metrics for such a class. We set $k = 7$ for the k-NN classifier and we set all the other hyperparameters consistently with the

[3]In our case, a client is a network telescope, which acts as a provider.

[4]Notice that in Word2Vec the produced embeddings are the learnable weights matrix between the input and the hidden layers.

TABLE I: Macro F1-Score and covered senders for the k-NN classifier applied on the host embeddings generated through different approaches.

	Local	Collaborative		Support
		Centralised	FL	
d_1	0.86	0.89	0.89	13 126
d_2	0.83	0.90	0.90	13 825
$d_1 \cup d_2$	–	0.89	0.89	16 560

TABLE II: Macro F1-Score and covered senders for the classifier applied on the host embeddings without incremental training.

	Local	Collaborative		Support
		Centralised	FL	
d_1	0.56	0.66	0.61	2 299
d_2	0.59	0.73	0.66	2 186
$d_1 \cup d_2$	–	0.65	0.60	2 839

validation reported in [2]. We partition our one-month dataset into 31 daily batches. For each batch, the FL approach involves only one round of aggregation. At each round, we update local models starting from the previous model (incremental update) for 1 epoch. We set the embedding size $E = 200$.

B. Two /24 network telescopes

In Table I we report the average F1-Score for the senders active in our collection. We consider a scenario where there are 2 providers d_1 and d_2, each with a /24 telescope. We report the metrics when considering

- *Local*: test only the senders active in each considered network telescope (d_1 and d_2); the embeddings are generated without collaboration;
- *Centralised*: test only the senders active in each considered network telescope; the embeddings are generated with the centralised approach;
- *FL*: test only the senders active in each considered network telescope; the embeddings are generated with the FL approach;
- *Joint*: the the whole set of senders observed in *both* the networks ($d_1 \cup d_2$); Embeddings are generated collaborating with centralised and FL approaches.

Performance benefit. Firstly, focus on the senders of the two network telescopes separately. Both the Centralised and FL approaches achieve good performance ($\geq$0.89 of average F1-Score in both providers) improving the local embeddings generation (F1-Score of 0.86 in d_1 and 0.83 in d_2). When focusing on the full set of observed senders ($d_1 \cup d_2$), the FL results are in line with the centralised ones, resulting in an average F1-Score of 0.89.

Coverage benefit. As at least 5 daily packets are needed to generate an embedding for a sender, the majority of the senders are deemed inactive. Since scanners and botnets target different telescopes with different intensities [4], some senders may be inactive in one telescope and be active in another one. Thus, building a single common model and vocabulary extends the embeddings to include more active senders independently from which telescope they are active, i.e. extending the support (last column). In fact, the federated setup allows thus the providers to build an embedding for about 3 000 additional senders compared to the ones locally observed for both d_1 and d_2, i.e., extending the support to include 14.77% and 12.25% more senders in d_1 and d_2 respectively.

In a nutshell, FL not only improves the embedding quality but it also allows to building of embeddings for more senders thus extending the support.

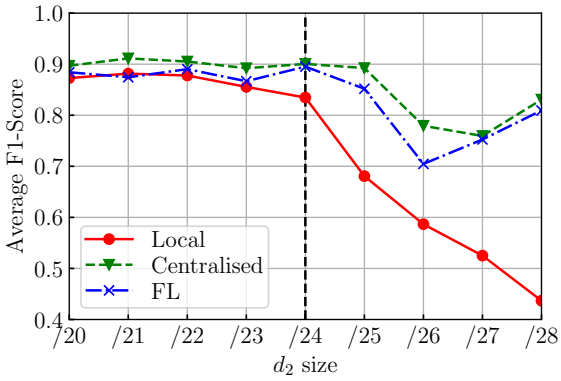


Fig. 3: Classification performance on d_2 of traffic observed by the /24 d_1 and a varying size d_2.

C. Benefit of incremental training

In Table II we report the performance when the incremental training is disabled. In this case, we train the embeddings from scratch only using the current batch of data, without benefitting from the previously accumulated knowledge represented by the previously learnt weights (trained over the 30 previous batches of data). With no incremental learning, each provider is only able to create embeddings for a few thousand senders. Also, the k-NN classifier performance is rather poor: the F1-Score drops to less than 0.6 in the local scenario. Collaboration cannot compensate the degradation in performance.

In a nutshell, the adoption of an incremental and continuous learning approach lets the embedder accumulate information. This produces a more informative representation and extends the coverage to more senders (including those that have been seen active in at least one batch of data in the past).

D. Two network telescopes of different sizes

In Figure 3 we evaluate the downstream classification task performance when the two telescopes are not equally sized. We consider the same /24 network telescope d_1 and vary the size of d_2 from /20 to /28. The vertical dashed line represents the case of Table I where the two telescopes have equal size.

Performance benefit. The benefits obtained by the larger network (leftmost part of Figure 3) show only marginal improvement compared to the local embeddings. Conversely, focusing on the benefits obtained by the smaller network, we observe substantial benefits from the collaborative approaches, resulting in a noticeable F1-score gain compared to the local

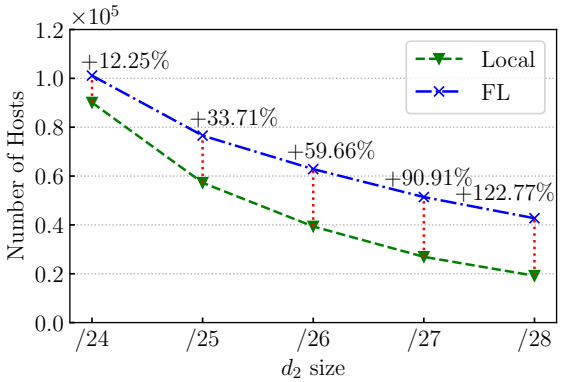


Fig. 4: Coverage of d_2 in the FL approach with a /24 d_1. Percentage shows the increase for FL.

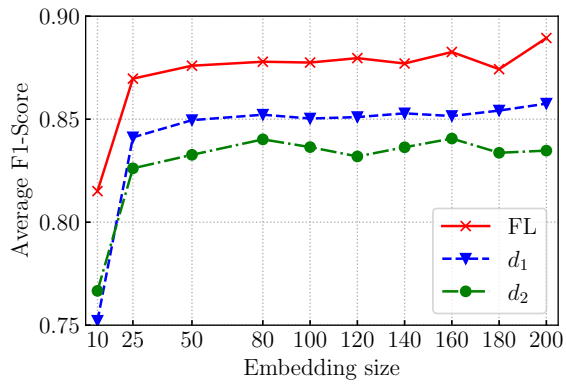


Fig. 6: Classification performance of traffic observed by the /24 d_1 and a /24 d_2, with different embedding sizes.

Fig. 5: Average F1-Score when embeddings are generated through the FL collaborative with more providers.

baseline. As illustrated in Figure 3, when the network telescope is extremely small (e.g./27 or /28) it heavily relies on the knowledge brought by the larger network telescope obtaining high-quality embeddings (F1-Score gain > 0.3 in d_2).

Coverage benefit. In Figure 4 we report the extended coverage due to the collaborative approaches. Overall, the sharing of information creates embeddings for more senders than the local approach, when collaborating with a network telescope of the same size, the coverage extends 12%. We highlight that smaller network telescopes strongly benefit from the federated setting. Thanks to the broader point of view of the /24 network, the embeddings coverage increases by 34% up to 123% when d_2 is a /28 network. The /24 d_1 telescope obtains similar benefits when federating its model with a larger d_2 telescope (not reported here for brevity).

E. Multiple network telescopes

Next, we evaluate the performance of the FL approach with multiple providers (i.e. network telescopes). We consider N network telescopes of the same size and report the resulting F1-Score in Figure 5. In this experiment, we extract all the network telescopes from the Brazilian network telescope by extracting equally spaced subnets. When all the providers observe a sufficient amount of traffic to generate meaningful embeddings locally (e.g./24 or /25 network telescopes) the collaborative improves the quality of embeddings as reflected by the improved F1-Score. Interestingly, /26 or smaller network telescopes do not accumulate enough data to produce robust embeddings. Their collaboration is less beneficial in this case.

V. Quantifying the Exchanged Information

Both the FL and centralised approaches require the providers to share information with the central server. In this section, we estimate the amount of exchanged data.

A. Centralised scenario

In the Centralised approach each provider d shares all its data with the server. This results in hundreds of MB being sent. Even just sharing necessary features (in our case source IP addresses, timestamps and destination ports) amounts to 33.5 MB of data to be sent on average for each batch for a /24 subnet. Sharing the raw packet trace would cost about 5GB of data being exchanged[5].

Considering our specific case, each provider could share only the corpus C_d (i.e. the sequence of senders as they target specific ports) to train the Word2Vec model in the central server. In this case, the volume of exchange data is proportional to the amount of observed traffic. Formally, $I_d^N = |C_d| \cdot s$, where s denotes the size of the identifier of an IP address (i.e. 32 bits). On the last day of our collection, the /24 network telescope d_1 observes ≈ 2.5 million packets, resulting in an exchange of $I_{d_1}^N = 5.3$ MB of data.

B. Federated learning approach

In the FL solution, each provider d transmits its vocabulary V_d and two embedding matrices sized $|V_d| \times E$ each[6]. Thus, the amount of exchanged information by provider d is $I_d^{FL} =$

[5]Notice that in certain scenarios, such as Distributed Denial of Service (DDoS) attacks, the large number of sent packets might increase significantly resulting in higher information exchange.

[6]The first is the actual embedding matrix $\mathbf{X}_d$ defined in Section III, while the second is used for negative sampling [19]

$|V_d| \cdot (2E \cdot f + s)$, where f denotes a floating-point number size (i.e. 64 bits). On the last day of the collection, network telescope d_1 observes ≈ 9 thousand senders. With $E = 200$, this results in an exchange of $I_{d_1}^{FL} = 27.8$ MB of data.

In [2] the embedding size is set to $E = 200$ through a sensitivity analysis, given the transmitted information can be substantially reduced through a small embedding size, here we investigate if FL allows to reduce the embeddings size.

We report in Figure 6 the impact of E on classification performance. In line with [2], larger embeddings can better represent the information leading to an improved performance (F1-Score > 0.82 with $E = 200$ for both local case and with FL). Overall, for embeddings whose size is in the $[50, 200]$ range, the impact of E becomes marginal and the FL approach leads to consistent F1-Score improvement. Conversely, embeddings smaller than $E = 50$ do not represent enough information, degrading the classification performance – e.g. F1-Score < 0.77 for local d_1 and d_2 embeddings with $E = 10$.

By reducing E, the benefit of collaborating is even more evident: With $E = 50$ only 7.0 MB of data is exchanged, yet achieving 0.88 of F1-score. Compared to the 5GB of raw traffic data, FL guarantees a significant reduction in the data exchange cost.

Notice that in our use case, the adoption of an FL approach introduces a marginal delay in the update of the embeddings (which finishes in one round) and thus on the classification task.

C. Privacy consideration

Notably, even though the Centralised approach only requires sending the preprocessed traffic, i.e. the corpus, to the server, it still entails sharing sensitive information.

In detail, the corpus reports (i) the actual amount of traffic directed at the provider's network telescope, (ii) the temporal co-occurrence of senders generating this traffic, (iii) unveils the presence of possible infected machines[7]. Furthermore, if the data are leaked to attackers, they will be able to locate the IP addresses of telescopes so that they can avoid these ranges in future activities, rendering network telescopes ineffective for their purpose.

Conversely, the embeddings shared through the FL approach are simply numerical representations of observed senders.

VI. Conclusions

In this paper, we presented an exploration of incremental federated learning solutions for generating self-supervised host embeddings from darknet traffic analysis. We embrace a 2-stage pipeline and extensively evaluate Centralised and FL collaborative solutions.

Overall, our work underscores the potential of FL and incremental learning as a valuable tool in the analysis of network traffic. Future works should encompass a more comprehensive exploration of FL adaptation in traffic analysis. Additionally, there is the possibility of extracting information from diverse sources beyond darknet traffic, such as honeypot data. Finally, the embeddings generated from different providers can find applications in other downstream tasks like clustering and anomaly detection.

[7]Because some viruses and worms involve the installation of backdoors that provide unfettered access to infected computers, telescope data may inadvertently advertise these vulnerable machines

Acknowledgment

This work was supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] M. Kallitsis, R. Prajapati, V. Honavar, D. Wu, and J. Yen, "Detecting and interpreting changes in scanning behavior in large network telescopes," *IEEE Transactions on Information Forensics and Security*, 2022.

[2] L. Gioacchini, L. Vassio, M. Mellia, I. Drago, Z. Houidi, and D. Rossi, "i-darkvec: Incremental embeddings for darknet traffic analysis," *ACM Transactions on Internet Technology*, 2023.

[3] C. Fachkha and M. Debbabi, "Darknet as a Source of Cyber Intelligence: Survey, Taxonomy, and Characterization," *Commun. Surveys Tuts.*, 2016.

[4] F. Soro, I. Drago, M. Trevisan, M. Mellia, J. Ceron, and J. J. Santanna, "Are darknets all the same? on darknet visibility for security monitoring," in *2019 IEEE LANMAN*, 2019.

[5] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and S. Philip, "Graph self-supervised learning: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[6] Z. Houidi, R. Azorin, M. Gallo, A. Finamore, and D. Rossi, "Towards a systematic multi-modal representation learning for network data," in *Proceedings of the ACM Workshop on Hot Topics in Networks*, 2022.

[7] M. Ring, A. Dallmann, D. Landes, and A. Hotho, "IP2Vec: Learning Similarities Between IP Addresses," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017.

[8] D. Cohen, Y. Mirsky, Y. Elovici, R. Puzis, M. Kamp, T. Martin, and A. Shabtai, "DANTE: A framework for mining and monitoring darknet traffic," in *Proceedings of the European Symposium on Research in Computer Security*, 2020.

[9] L. Gioacchini, L. Vassio, M. Mellia, I. Drago, Z. Houidi, and D. Rossi, "DarkVec: automatic analysis of darknet traffic with word embeddings," in *Proceedings of CoNEXT*, 2021.

[10] CAIDA, "The ucsd network telescope," https://www.caida.org/projects/network_telescope, 2024.

[11] C. Han, J. Takeuchi, T. Takahashi, and D. Inoue, "Dark-tracer: Early detection framework for malware activity based on anomalous spatiotemporal patterns," *IEEE Access*, 2022.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017.

[13] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion detection for wireless edge networks based on federated learning," *IEEE Access*, 2020.

[14] H. Fereidooni, A. Dmitrienko, P. Rieger, M. Miettinen, A.-R. Sadeghi, and F. Madlener, "Fedcri: Federated mobile cyber-risk intelligence," in *Network and Distributed Systems Security (NDSS) Symposium*, 2022.

[15] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal*, 2022.

[16] V. Pourahmadi, H. A. Alameddine, M. A. Salahuddin, and R. Boutaba, "Spotting anomalies at the edge: Outlier exposure-based cross-silo federated learning for ddos detection," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[17] Y.-W. Chang, H.-Y. Chen, C. Han, T. Morikawa, T. Takahashi, and T.-N. Lin, "Finish: Efficient and scalable nmf-based federated learning for detecting malware activities," *IEEE Transactions on Emerging Topics in Computing*, 2023.

[18] J. Ceron, K. Steding-Jessen, C. Hoepers, L. Granville, and C. Margi, "Improving iot botnet investigation using an adaptive network layer," *Sensors*, 2019.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.