Towards Reliable Network Entity Tracking Using Behavioural Bag of Words Representation

Jaroslav Hlaváč[1,2], Martin Kopp[1], Michael Polák[1], and Jan Kohout[1]

[1] Cisco Systems, Cognitive Research Team in Prague
[2] Faculty of Information Technology, Czech Technical University in Prague

Abstract: In this paper, we study the problem of entity identification and tracking in the domain of network security. The ability to uniquely identify and track network entities in time is essential for network behavioural analytics. Our approach leverages the Bag of Words (BoW) representation, enabling us to build representations from many different features from multiple data sources.

However, normalisation methods traditionally used with BoW in other application domains (e.g. tf-idf, stop words) do not work well with network data as they are not designed to capture behavioral patterns. Some features, such as common networks servers (e.g. google.com, update.microsoft.com), or executed binaries (e.g. web browsers), are often too frequent but still valuable behavioural indicators. In order to capture important long-term patterns in entities' behaviour, we introduce time-aware normalisation of the BoW representations.

We compare different representations for device tracking on real network telemetry. Our results show that using multiple data sources significantly improves entity tracking, especially when combined with proposed time-aware normalisation.

Keywords: bag of words, entity tracking, network behavioural analytics

1 Introduction

The ability to identify and track network entities (devices, users, etc.) is crucial for user and entity behavioural analytics (UEBA), which is the application domain of our paper. UEBA systems usually create a representation of the normal behaviour of an entity and then look for abnormal activity in the subsequent network communication. In general, an entity representation is a numerical vector in a latent space that describes real-world objects, such as words or movies, their relationships, and behaviour. Ideally, the semantic similarity in the input space is captured in the latent space, meaning that similar input objects (words, movies) are close to each other in the latent space. In our case, the input objects are network devices and users.

We are looking for a representation that would serve as a unique fingerprint for each device/user, and differences between them would correspond to differences in their behaviour. A straightforward way to represent user or device behaviour on the network is by sets of features, such as visited application servers, web domains, or used programs. Unfortunately, such representation does not form a metric space as it supports only pair-wise similarity comparisons (see [7] for a more detailed explanation). Furthermore, such representation is inconvenient to work with, and set operations are computationally expensive. The representation in the latent space helps to overcome these issues.

There are other challenges that an ideal representation should respect. The entities (users and devices) change in time; therefore, their representation moves in the latent space. Tracking this movement opens new possibilities for anomaly detection. Any significant shift in latent space indicates a sudden change in user/device behaviour and may be worth reporting as an anomaly. Clustering the representations could help to discover groups of similarly behaving entities. Finding that an entity has changed or is frequently changing its group can be treated as an anomaly.

In this paper, we are focusing on the first problem, which is user/device tracking. Uniquely identifying and tracking any network device is the first step of all the above-mentioned use cases. We compare device representations based on the Bag of Words (BoW) built on top of multiple features from different data sources as well as their combination. BoW is a universal approach that allows explaining the differences in representations directly from feature vectors.

Furthermore, we introduce the time-aware normalisation of the BoW representations to reduce the influence of the most common network servers, binaries, etc. We compare the representations on a week of telemetry gathered in a real company network.

The rest of the paper is organised as follows. The next section covers the related work in the field of entity representation. Section 3 formally describes the entity representation task, followed by the experimental evaluation in Section 4. The Section 5 concludes the paper.

2 Related work

There are two major challenges that need to be solved in the task of **time-aware tracking** of network entities. Firstly, representations need to change in time as the behaviour of the entity evolves. Secondly, the change in net-

work entity behaviour needs to be easily **explainable** from the deviation in the representation vector.

We are not aware of any prior research in finding representations from multi-modal data in the area of network security. However, the problem of finding entity representations is actively researched in other domains, such as healthcare [3], graph node classification [5, 6], natural language processing [13], recommender systems [2] and others.

Recommender systems use **time-aware representation** to successfully predict what item the user might be interested in during the next interaction with the application, e.g. the next movie to watch [4] or next item to buy [9]. Using recurrent neural networks (RNNs) that take user-item transactions as an input (such as JODIE [9]) has recently proven to be promising in tracking and predicting the trajectories of both users and items. However, in our use case in network security, they suffer from a lack of explainability. We need to track trajectories of the entity, but we also need to attribute the change in behaviour to a specific feature or set of transactions that caused the deviation. This allows faster investigation of a potential security incident.

The explainability problem can be directly solved by the Bag of Words representation. It brings the possibility to tie the (dis)similarity of two objects with concrete features. BoW is a well-known and effective approach originally used for document classification but also in other domains like image recognition [19, 12] and NLP [8]. It suffers from two weaknesses: sparsity of the resulting representations and the inability to capture the semantics of the represented entity.

The sparsity in the latent space is often tackled by entirely removing the most common features and using only top N most frequent features from the remaining vocabulary. In our use case, this approach is unfeasible as both high and low frequented features are needed to capture the behavior of an entity.

In [10] the semantics are captured by dividing the image into subsections and applying the BoW representation on them. We also divide the traffic into smaller parts. But as our data are time series and not images, we exploit temporal (time windows) rather than spatial vicinity.

The problem of uniquely identifying users is also being studied in user-computer interaction. Passive monitoring of user actions is used to identify the user. For example, [15] treats keyboard strokes and mouse movement as a biometric means of authentication. In [1], they were able to uniquely identify users by collecting data from browser sessions. Our problem is more complex, as we do not have access to the direct interaction of the human user with the computer nor to active probing. Our only source of data is passive log access. Even distinguishing between human and machine generated components of network traffic is a difficult task.

Lastly, related problem is studied in named-entity recognition [14, 11]. Named entity recognition focuses on finding tokens that identify entities of predefined categories (names, currencies, etc.) in the textual or similar data. The methods often rely on some contextual knowledge from surrounding words or sentences. However, we can hardly rely on such context in network data as it can be scattered over several hundred or even thousands of logs. Also, the entities we are tracking are users and devices that exhibit complex and very dynamic behaviour that is often changing in time. Therefore, methods from named-entity recognition are not directly applicable.

3 Entity Representation

The representation of an entity is the result of a mapping $e\colon X \to L$ from a general Cartesian feature space X to a latent space L where a numerical vector represents each entity. We assume that there exists a similarity function:

$$\mathrm{sim} : L \times L \to [0,1]$$

in the latent space L. This similarity function fulfils the standard requirements of symmetry and identity of indiscernibles.

The goal is to find such mapping that would create a time-aware behavioural fingerprint of an entity. Formally, we define following requirements for the mapping:

- **Requirement 1**: Representations of an entity in two subsequent time periods should be similar to each other. This can be expressed by following formula for average self-similarity:

$$r_1 = \frac{1}{N}\sum_{i}^{N} \mathrm{sim}\left(e(x_{it_1}), e(x_{it_2})\right), \tag{1}$$

  where N is the number of entities in the set, e is the mapping from raw feature space X to latent space L, $x_i \in X$ is the entity representation in X, and t_1, t_2 are the consecutive time periods.

- **Requirement 2**: Different entities need to be dissimilar and distinguishable by their representation. This can be expressed by the following formula for average dissimilarity between different entities:

$$r_2 = \frac{2}{N(N-1)}\sum_{i}^{N}\sum_{j}^{i-1}\left(1 - \mathrm{sim}(e(x_{it}), e(x_{jt}))\right) \tag{2}$$

  where N is the number of entities in the set, e is the mapping from raw feature space X to latent space L, and $x_i, x_j \in X$ are different entities.

With the requirements above and a given similarity function sim we want to find the mapping e which maximises both requirements r_1,r_2, e.g., in a form of their weighted sum.

3.1 Bag-of-Words Representation

In this work we are using **bag of words (BoW)** [16] representation as a baseline for further work. BoW is an information retrieval technique originating in document classification. It is used to represent a document in a vector space by computing the number of term occurrences, discarding the document's structure. A term is usually a word or n-gram. The dimension of the representation space is determined by the number of unique terms (called vocabulary) in the set of compared documents.

Having network flows and endpoint logs at disposal, the bag is constructed from all values (terms) observed in one feature in a given time period, i.e. all executable hashes used by a device in one day (treated as a "document".) would be added to the bag. The vocabulary would then be all the hashes used by the devices in the network in an extended time window (e.g. day, week).

Using only counts of occurrences does not work well for many of the features as usually few values occur significantly more often than others. Thus, all vectors may look similar because of this frequent feature. In the case of executable hashes, this could be the hash of Google Chrome, as it is the most common browser. Therefore, tf-idf (term frequency - inverse document frequency) [17] is used to weight the vector by the amount of information each term brings. If the term is very common, the idf value is small, reducing the impact of the term in the resulting vector.

The frequent features can have several orders of magnitude more occurrences than the other features. In the document classification problems, the most frequented words (is, are, with, the, a, an etc.) can be removed from the vocabulary. It is not the case in the network and endpoint telemetry as the most frequented terms can change (e.g. updating a program changes the executable's file hash). Or they can contain valuable information for entity identification (e.g. the most frequented autonomous systems (AS) contacted by Windows machines are maintained by Microsoft, distinguishing them from Linux machines).

According to our experiment, using tf-idf to re-weight features is not enough. Therefore, we utilised **time window bag of words (tw-BoW)** representation. In tw-BoW, each feature is counted only once for each time window it occurred in (e.g. for ports, no matter how many times in a time window device accessed port 80, it counts as only one occurrence). This creates a constraint on the maximal value of each vector component (e.g. 288 for a representation of one day split into 5-minute windows). Smoothing the vector by this method enables less significant values for a given feature to have a bigger impact on the final vector. Otherwise, the most frequent features could overshadow others even after tf-idf smoothing.

4 Experiments

This section covers experiments with different entity representations. The main goal was to compare mappings from raw feature space to latent vector space on the use case of tracking entities in the network over time.

The endpoint client IDs were used as labels for the purpose of device tracking.

Two possible mapping approaches based on the BoW method explained in Section 3.1 are compared in the experiments. The first approach is classic BoW representation, where feature frequencies are computed from all term occurrences (e.g. for ports, each access on destination port 80 counts as one occurrence), tf-idf is later used to re-weight each feature according to the frequencies observed in the network.

The second approach is the time window BoW representation (again weighted by tf-idf), where each feature counts only once for each time window it occurred in. For this experiment, we used 5-minute time windows. Therefore each vector component ranges from 0 to 288 (number of 5-minute windows in 24 hours).

4.1 Evaluation

The representations were evaluated according to their ability to track the device in time in the latent space. This evaluation criterion is formalised by the requirements 1 and 2 in Section 3. To evaluate the quality of a mapping, similarities between all devices appearing in one day, and all devices appearing in the other day were computed. Total of $N * M$ similarities were computed for every two days, where N is the number of devices in the first day and M is the number of devices in the second day. The M similarities in each row i of the matrix were ranked according to:

$$\text{rank}_i = 1 + |\{j | \text{sim}(e(x_{it_1}), e(x_{jt_2})) > \text{sim}(e(x_{it_1}), e(x_{it_2})\}|, \tag{3}$$

where sim is a similarity measure, $e(x_{it_1}), e(x_{jt_2}) \in L$ are representations of different devices x_i, x_j in the latent space L and times t_1, t_2 are two consecutive days.

The following metrics were used to compare the quality of the representations:

- **Mean rank R** in which each device representation appeared in the second day:

$$R = \frac{\sum_i^M \text{rank}_i}{M}, \tag{4}$$

  where M is the number of devices in the second day and rank_i is the rank from Equation 3. The lower mean rank, the better the representation. In the best case the mean rank would be 1, allowing to precisely track all devices over time.

- **Percentage of precise hits A** is defined as:

$$A = \frac{\sum_i^M \mathbb{I}[\text{rank}_i = 1]}{M} \tag{5}$$

where M is the number of devices in the second day, $\mathbb{I}$ is the indicator function which is 1 if the rank is equal to 1 and zero otherwise and rank_i is the rank from Equation 3.

The value of A shows the portion of devices that could be uniquely identified. If there are multiple devices tied with the same highest similarity, it does not count as precise hit, because the device cannot be identified uniquely (cf. Requirement 2 in Section 3). For example, when two devices access the same set of autonomous systems during the day, they both correctly appear at the first rank. However, they cannot be differentiated based on ASN.

- **Cumulative distribution function** (CDF) of the device appearing on rank N or lower for each device:

$$f(x) = P(\text{rank}_i \leq x) \tag{6}$$

where the right-hand side represents the probability of randomly selected device x having higher rank than rank_i. This metric is useful for comparison between different algorithms.

These metrics enable comparison of both BoW and time window BoW approaches as well as comparing the different features used to create them.

4.2 Experimental Setup

The experiments were performed on a week of real telemetry (Jan 11 to Jan 18, 2021) from a corporate network. The telemetry was collected both on the endpoint devices and network proxies. Table 1 shows the numbers of devices present in the network for each day in the week. The difference between numbers of devices seen at the endpoint and in the network is mainly caused by endpoint devices that communicate only within the private network. Therefore, the traffic was not observed on a proxy. A significant drop in device number can be observed during the weekend.

We used three different features, listed in Table 2, to test the viability of the BoW approach. Private destination IPs are addresses falling into ranges defined in RFC1918 [18]. Private IP address was chosen because it is one of the few that captures behaviour on the internal network. They were collected on the endpoint as network proxies usually do not handle internal traffic.

The file hash is a string uniquely identifying a file. Files that were created, opened or executed on the endpoint are supplied to a hashing function to compute the hash.

Autonomous system number comes from enriching the network telemetry with information from a GeoIP database. It is expected that a single network in one location will mostly communicate with several autonomous systems. The dominant autonomous systems will be similar for each device in the network. The experiment is designed to test whether the remaining less frequent ASNs can serve to distinguish devices.

Date	#Devices Endpoint Data	#Devices Network Data
Jan 11 (Mon)	1764	1057
Jan 12 (Tue)	1802	1065
Jan 13 (Wed)	1796	1087
Jan 14 (Thu)	1790	1068
Jan 15 (Fri)	1754	1056
Jan 16 (Sat)	1667	979
Jan 17 (Sun)	1154	714
Jan 18 (Mon)	1801	1064

Table 1: Number of devices observed in different telemetries on the network

Feature	Description	Data Source
Dst IP	Private range IP address	Endpoint
File Hash	Hash of the inspected file	Endpoint
ASN	Autonomous System Number for destination IP	Network

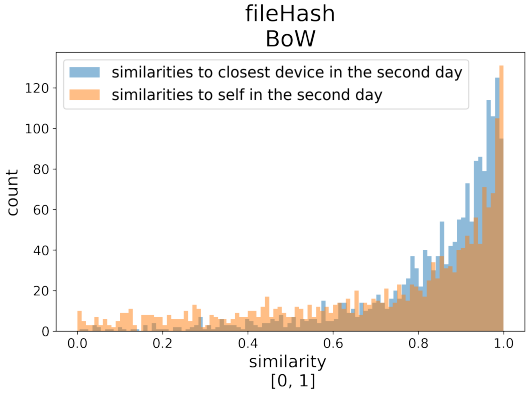
Table 2: Features tested for device representations with source telemetry for individual feature.

The representations were created for one feature at a time using the BoW and time window BoW approach covered in Section 3.1. One day (24 hours) period was selected to create a representation, assuming that it contains most of the regular behavioural routines of the device and is small enough to detect the change in behaviour as soon as possible. The dimensions of the latent spaces (defined by the vocabulary size) changed between different days. They were ~ 3500 for `dstIpPrivate`, and ~ 12000 for `fileHash`, and ~ 850 for `autonomusSystemNumber` changing slightly every day.

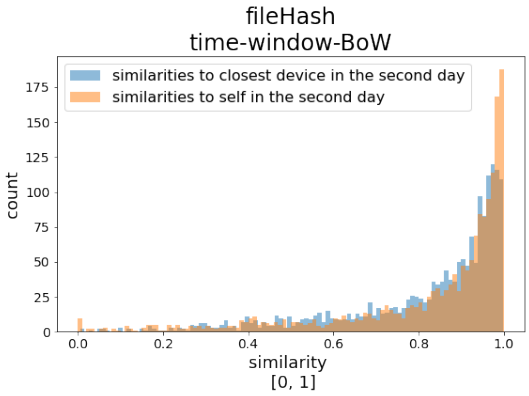
Two representations (one for each day) were created every two consecutive days to test the device tracking in time. The vocabulary used for BoW mapping contains all terms (observed feature values) that occurred during these two days. A days representation was created for each device by counting feature occurrences and re-weighting the resulting vector by tf-idf. Cosine similarity between representations from consecutive days was used to evaluate the quality of representations. This process was repeated for each day of the week.

4.3 Results

Figure 1 shows similarity distributions to self and the most similar device using the `fileHash` feature. The plotted histogram represents similarities of device representations



(a) fileHash: BoW



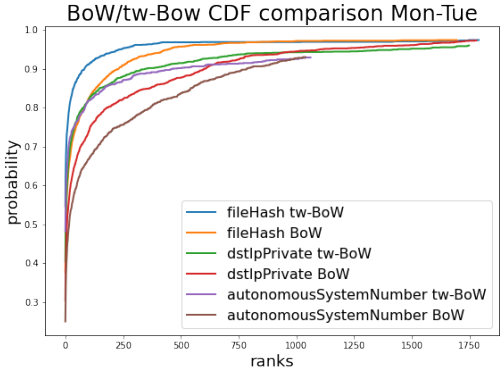
(b) fileHash: tw-BoW

Figure 1: Histograms of similarity distributions for device representations between Monday January 11 and Tuesday January 12, 2021.

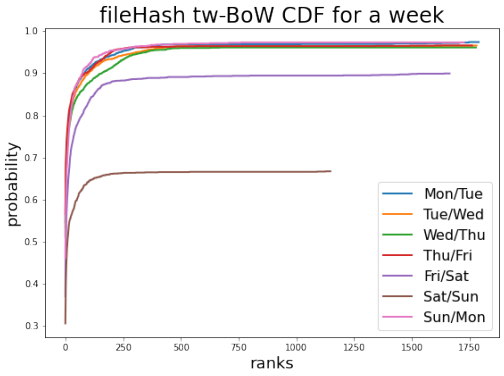
Feature	Mapping	Mean Sim to Self	Mean Sim to Closest
dstIp	BoW	0.78 ± 0.33	0.86 ± 0.25
	tw-BoW	0.78 ± 0.26	0.81 ± 0.22
file hash	BoW	0.73 ± 0.27	0.84 ± 0.18
	tw-BoW	0.82 ± 0.22	0.82 ± 0.2
ASN	BoW	0.85 ± 0.21	0.95 ± 0.09
	tw-BoW	0.91 ± 0.15	0.95 ± 0.09

Table 3: Mean similarities with standard deviations to self and to the closest device between two days.

from Monday, January 11, and Tuesday, January 12. For other days the distributions are similar. The orange colour depicts the similarity to self on the second day. Better representation of devices can be seen from the number of orange bars that are higher than the blue ones. Mean similarities to self and the closest other device are listed in Table 3. By looking at the results, it is clear that using tw-BoW increases mean similarity to self in relation to mean similarity to other devices. However, mean similarities to the closest device are still higher, indicating that individual features are not enough for accurate device tracking.



(a) All representations with the CDFs of rank for probability of the device being within the top N ranks of similarities in the next day.



(b) CDF of rank for `fileHash` in different days of the week. The quality of representations deteriorates significantly over the weekend as the number of devices drops significantly.

Figure 2: Comparing CDFs of rank of devices appearing on N-th rank between different features (a) and different days of the week (b).

Figure 1 visualises what that means for the `fileHash` feature.

To compare the ability to track devices in time, cumulative distribution functions from Equation 6 are plotted in Figure 2a. Bigger area under the CDF indicates better representation. tw-BoW representations outperform the raw BoW approach for all the tested features. File hash shows the best results, with 75% of devices being in the top ten ranks in the second-day representations. The autonomous system number performs the worst of the three features. This indicates that infrequently accessed autonomous systems are not enough to differentiate between devices. Private destination IP address slightly outperforms the ASN feature. After inspecting the data, most private network communications were to several addresses that belong to load-balanced servers.

Complete results averaged over the whole week are listed in Table 4. Best values for each category are highlighted in bold. In all three tested features, the time win-

dow BoW representation has shown better results, and file hash proved to be the best feature for device tracking.

Feature	Mapping	Mean Rank	Precise Hit
dstIp	BoW	142.55 ± 21.96	0.27 ± 0.06
	tw-BoW	86.16 ± 15.20	0.34 ± 0.07
file hash	BoW	55.82 ± 10.98	0.35 ± 0.06
	tw-BoW	$\mathbf{25.65 \pm 4.99}$	$\mathbf{0.47 \pm 0.10}$
ASN	BoW	119.71 ± 21.49	0.22 ± 0.01
	tw-BoW	49.30 ± 10.20	0.42 ± 0.05

Table 4: Comparison of mean rank and precise hits of BoW and tw-BoW representations for each feature averaged over the whole week. Mean rank shows the efficiency of tracking individual users (the lower, the better). Precise hits ratio shows the percentage of devices that were tracked accurately over time.

Figure 2b shows how the CDFs change over the course of the week for `fileHash`. CDFs during workdays are quite stable. During the weekend, the behaviour changes, and probabilities are significantly lower. The drop from Friday to Saturday is smaller than from Saturday to Sunday. This is probably caused by the time shift as the data is timestamped in GMT but the network is located in GMT-6 timezone. The probabilities are high again from Sunday to Monday because the devices running over the weekend are probably servers. Their behaviour is maintained over time. The drop in probability over the weekend could be mitigated by tracking devices over workdays separately from weekends.

The last experiment performed was the use of combined representations for device tracking. The already precomputed representations were used to compute similarity matrices to all devices and then the three similarities were averaged:

$$\mathrm{sim}_{avg}(x_i, x_j) = \frac{1}{|M|} \sum_{m \in M} \mathrm{sim}_m(x_i, x_j), \qquad (7)$$

where $M = \{dstIp, fileHash, ASN\}$ is a set enumerating similarities for different features and x_i, x_j are the devices compared.

Only devices that were present in all three telemetries were used in this experiment, which significantly reduces the number of devices in the dataset. Figure 3 shows the CDFs for ranks using common devices. Using the average similarity shows an improvement, increasing the precise hit ratio significantly. Concrete numbers can be found in Table 5, together with results for individual features on the reduced dataset.

Lastly, to visualise the difference between all approaches, Figure 4 shows confusion matrices for 50 randomly selected devices. Lighter tile colour means higher similarity. From Figure 4a private destination IP address does not seem to be a good feature for individual user tracking. However, groups of devices behaving very similarly might be harvested from the data. Looking at the raw

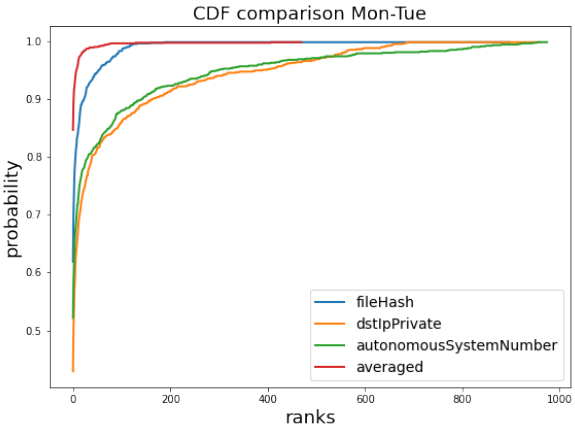


Figure 3: CDFs for common subset of devices together with averaged similarity over all features using Equation 7 with tw-BoW representations used as an input.

Feature	Mean Rank	Precise Hit
average	**3.32**	**0.85**
fileHash	9.56	0.62
dstIP	53.00	0.43
ASN	50.73	0.50

Table 5: Results of using average similarity for computing similarity between a subset devices present in both network and endpoint telemetry between Jan 11 and Jan 12, 2021.

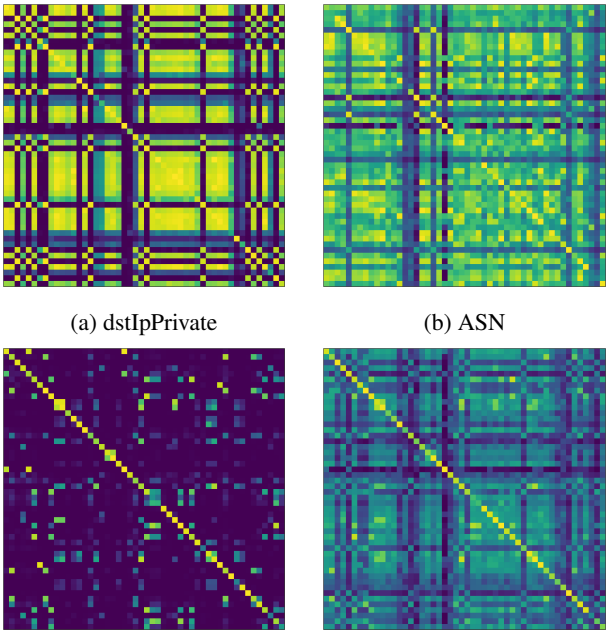
data has revealed that the similarity comes mainly from few common IP addresses. These addresses are LDAP and HTTP servers in the network. However, as these servers use load balancing, the clusters are not stable in time as devices communicate to different IP addresses. Figure 4c differentiates most of the devices well, with several exceptions (very light squares off the diagonal). The averaged similarities shown in Figure 4d reduce the impact of these exceptions, which corresponds with the results in Table 5.

4.4 Discussion

Surprisingly, the time window BoW representation has proven to be quite effective for device tracking while using only three easily interpretable features from different modalities. More features and modalities can be added for further improvement.

Even though file hashes alone show promising results, they do not enable to confidently track the device over time. Averaging the similarities from different feature representations significantly increases the number of precisely tracked devices. The downside is that both network and endpoint telemetries have to be present. This can be improved in the future by combining the representations from different features even if one feature is missing.

The results also indicate that the behaviour of most of the network devices differs significantly between work-



(a) dstIpPrivate (b) ASN

(c) fileHash (d) average

Figure 4: Confusion matrices for 50 randomly selected devices for different features.

days and weekends. Creating separate representations for weekdays and weekends could yield better results in overall tracking.

5 Conclusion

In this paper, we studied the problem of user/device identification and tracking. Uniquely identifying and tracking any network device is the first step to proper user and entity behavioural analytics (UEBA). As a baseline, we tested device representations based on Bag of Words (BoW) built from multiple features from different data sources. Our experiments on real-world data showed that combining these features is beneficial for device identification and tracking.

Furthermore, we introduced the time-aware normalisation of the BoW representations. It reduces the influence of the most common values in each data source (network servers, binaries, autonomous systems) and significantly improves unique user/device identification accuracy.

In the future, we would like to focus on learning lower-dimensional representations that would still capture the entity behaviour and would be adaptable in time.

References

[1] Károly Boda, Ádám Máté Földes, Gábor György Gulyás, and Sándor Imre. User tracking on the web via cross-browser fingerprinting. In *Nordic conference on secure it systems*, pages 31–46. Springer, 2011.

[2] Hung-Hsuan Chen. Behavior2vec: Generating distributed representations of users' behaviors on products for recommender systems. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(4):1–20, 2018.

[3] Sajad Darabi, Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. Taper: Time-aware patient ehr representation. *IEEE journal of biomedical and health informatics*, 24(11):3268–3275, 2020.

[4] Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2015.

[5] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[6] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[7] Jaroslav Hlaváč, Martin Kopp, and Jan Kohout. Cluster representatives selection in non-metric spaces for nearest prototype classification. *arXiv preprint arXiv:2107.01345*, 2021.

[8] Monisha Kanakaraj and Ram Mohana Reddy Guddeti. Nlp based sentiment analysis on twitter data using ensemble classifiers. In *2015 3Rd international conference on signal processing, communication and networking (ICSCN)*, pages 1–5. IEEE, 2015.

[9] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278, 2019.

[10] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[11] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[12] Teng Li, Tao Mei, In-So Kweon, and Xian-Sheng Hua. Contextual bag-of-words for visual categorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(4):381–392, 2010.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[14] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[15] Piotr Panasiuk, Maciej Szymkowski, Marcin Dąbrowski, and Khalid Saeed. A multimodal biometric user identification system based on keystroke dynamics and mouse movements. In *IFIP International Conference on Computer Information Systems and Industrial Management*, pages 672–681. Springer, 2016.

[16] Nikolaos Passalis and Anastasios Tefas. Entropy optimized feature-based bag-of-words representation for information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1664–1677, 2016.

[17] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[18] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, IETF, February 1996.

[19] Chih-Fong Tsai. Bag-of-words representation in image annotation: A review. *International Scholarly Research Notices*, 2012, 2012.