Defending a Music Recommender Against Hubness-Based Adversarial Attacks

Katharina Hoedt[1] **Arthur Flexer**[1] **Gerhard Widmer**[1,2]
[1] Johannes Kepler University Linz, Austria
[2] LIT AI Lab, Linz Institute of Technology, Austria
katharina.hoedt@jku.at arthur.flexer@jku.at

ABSTRACT

Adversarial attacks can drastically degrade performance of recommenders and other machine learning systems, resulting in an increased demand for defence mechanisms. We present a new line of defence against attacks which exploit a vulnerability of recommenders that operate in high dimensional data spaces (the so-called *hubness problem*). We use a global data scaling method, namely Mutual Proximity (MP), to defend a real-world music recommender which previously was susceptible to attacks that inflated the number of times a particular song was recommended. We find that using MP as a defence greatly increases robustness of the recommender against a range of attacks, with success rates of attacks around 44% (before defence) dropping to less than 6% (after defence). Additionally, adversarial examples still able to fool the defended system do so at the price of noticeably lower audio quality as shown by a decreased average SNR.

1. INTRODUCTION

Adversarial examples were previously reported in various fields of application (cf. [1–3]) as small perturbations of input data that drastically change the performance of machine learning systems. Since then, numerous attempts to make systems more robust and *defend* them against these attacks have been made (cf. [4–6]).

In previous work [7], a successful attack on the music recommender FM4 soundpark [8] was presented, inflating the number of times (perturbed) songs were recommended by the system. The paper failed, however, to provide an outlook on how this attack could be weakened and did not present a method to defend against the attack. The proposed attack exploited *hubness*, a problem of learning in high dimensions that leads to some songs being recommended very often and other songs being never recommended [9]. As the attack amplifies the negative effect of hubness on the recommendation of songs, finding a defence against this kind of attacks would contribute positively towards the fairness of a recommender system.

To explain the existence of hub points, it was previously shown that for any high but still finite dimensionality, some points are expected to be closer to the center of all data (or local center in case of multimodal data) than other points and are at the same time closer, on average, to all other points [10]. Such hub points appear in nearest neighbour lists of many other points, resulting in asymmetric neighbour relations: a hub y is the nearest neighbour of x, but the nearest neighbour of the hub y is another point a ($a \neq x$). One approach to repair asymmetric neighbourhood relations and hence mitigate the hubness problem is Mutual Proximity (MP) [11], which globally scales distances (i.e., considers neighbourhood information of *all* objects), and transforms the distance between two objects into a measure that captures how similar the neighbourhoods of these two objects are.

In this work, we aim at finding a defence against adversarial attacks that exploit the hubness issue present in various recommender systems. Our contribution in this paper is threefold: (i) for the first time we utilise the hubness-reduction method MP as a defence against adversarial attacks; (ii) in order to create a more difficult defence scenario for MP, we additionally incorporate knowledge of the defence into a modified adversarial attack; and (iii) we also investigate MP as a post-hoc defence, i.e., using MP to post-process recommendations issued by an attacked but undefended system.

2. RELATED WORK

Hubness was first described in Music Information Retrieval (MIR) [12], but is now acknowledged as another aspect of the curse of dimensionality, and hence a general machine learning problem [10]. It was not only demonstrated to be a relevant problem in audio-based recommender systems [8, 9] (which are our focus here), but also in recommenders based on collaborative filtering [13, 14] as well as in general multimedia retrieval [15].

To reduce the effect of the hubness phenomenon, various hubness mitigation techniques were previously proposed. Feldbauer and Flexer differentiate between methods that use *dimensionality reduction*, centring-based methods that *reduce spatial centrality*, methods that aim at *repairing asymmetric relations* (e.g., MP), or using entirely *different distance measures* [16]. We look at MP in particular in this work, as it was previously applied to data of the FM4 soundpark [9], has been shown to be very effective at reducing hubness in diverse datasets [16] and is applicable even for large amounts of data with efficient approximations [17].

Defences against adversarial attacks can broadly be di-

vided into approaches in which adversaries are either *reduced* or *detected* (cf. [4]) in order to make a system more robust [18]. One of the most notable approaches is adversarial training [6], where adversarial examples are included in the training procedure of a system. As this is a complex task due to the necessity of computing adversarial examples during training time, multiple variants of this defence were proposed more recently (e.g., [19, 20]) with a focus on efficiency.

Adversarial defence methods are generally viewed to be at an arms race with adversarial attacks, as a lot of defences are only considered purposeful until a new attack is proposed which the method fails to defend against (cf. [18]). In this work, we therefore try to apply a method for defending against adversaries that does not require full knowledge of the specific attack; the one limitation we work with, however, is that the attack needs to exploit the hubness phenomenon. Note also that the real-world recommender that was attacked in previous work [7] would not permit adversarial training as a defence method due to the nature of the system (cf. section 4.1), as no training in the sense of learning model-parameters is performed.

In this work, we build upon two approaches that were previously published; first, we use the attack scenario proposed in [7] in order to provide the setting for our defence. Secondly, we apply the hubness-reduction method introduced in [9] and investigate its suitability as a defence method against adversarial examples. This new application is investigated in this work for the first time; in addition to that, we introduce an adapted attack which incorporates knowledge of MP and is hence equipped with more knowledge about the defence to test the boundaries of the proposed defence method.

3. DATA

The data we use in this work consists of 5,000 songs from the FM4 Soundpark [1] , a music discovery site that provides a platform where (lesser known) artists can upload and present their music free of charge. Website visitors can listen to and download all the music at no cost. Songs are generally assigned to one or two out of six different genres [8]. The songs have a duration of at least 2 and a maximum of 22.5 minutes, with an average of 4.1 minutes.

4. METHODS

4.1 The Music Recommender

The system we adapt in this work is the audio-based recommender system integrated in the FM4 Soundpark [2] [8], which was previously attacked in related work [7]. To prepare the data for the recommender system, every audio file is converted to mono and re-sampled to 22.05 kHz. The central two minutes of a song are further transformed to Mel Frequency Cepstrum Coefficients (MFCCs) by applying a Short-Time Fourier-Transform (Hann window with a window size of 1,024 and a hop size of 512), a transformation to Mel-scale and finally using a discrete cosine transform to compress the results to 20 MFCCs.

After the pre-processing step every song is represented by a multivariate Gaussian, which is estimated on the respective MFCCs [8]. To approximate the similarity of two songs in the next step, the Gaussians describing them are used to compute a symmetrised Kullback-Leibler (KL) divergence (cf. [8]), i.e.,

$$D_{\mathrm{SKL}}(\mathcal{G}_x, \mathcal{G}_t) = \frac{\mathrm{KL}(\mathcal{G}_x||\mathcal{G}_t) + \mathrm{KL}(\mathcal{G}_t||\mathcal{G}_x)}{2}. \quad (1)$$

Here $\mathcal{G}_x$ denotes the Gaussian computed to represent a song x and KL is the KL divergence of two Gaussian distributions. The symmetrised KL divergence allows us to determine the k nearest neighbours of any song using D_{SKL} as a distance measure. A recommendation for a particular song consists of these k nearest neighbours, where k is set to 5 as in the real-world recommender [8].

Note that for this recommender, adding a new song x' requires the computation of its Gaussian $\mathcal{G}_{x'}$, as well as obtaining the symmetric KL divergence between $\mathcal{G}_{x'}$ and all other Gaussians in the catalogue. This also means that for our experiments we can pre-compute these elements without loss of generality (cf. [11]).

4.2 Hubness

The music recommender we look at in this work suffers from the consequences of hubness, leading to over a third of the songs in the catalogue never being recommended [9]. So-called *hubs* are often among the nearest neighbours of songs and are hence frequently recommended, leaving no room for *anti-hubs* in nearest neighbour lists, and thus anti-hubs are never recommended. Hubs are usually determined by their *k-occurrence* O^k, which is a measure describing the number of times a particular object (here: song) is within the k nearest neighbours of all remaining objects that are part of a dataset. In what follows, we use a *hub-size* of $5k$, i.e., for a song to be considered a hub it has to have a k-occurrence $O^k \geq 5k$. Note that the mean O^k across all objects is equal k, with O^k significantly bigger than k indicating a hub. Anti-hub songs never occur within the nearest neighbours of other songs, meaning they have a k-occurrence $O^k = 0$; all remaining songs, with $0 < O^k < 5k$, are *normal* songs [9]. Note once again that we let $k = 5$ in our experiments, resulting in a hub-size that equals 25.

4.3 Mutual Proximity

Mutual Proximity was previously proposed as a global scaling method improving the negative effect of hubness by repairing asymmetries of the neighbourhood relation between any two objects in a dataset [11]. The main idea is to transform distances to a likelihood that one object x is the nearest neighbour to another object t (given the distribution of all distances to object t), and to combine it with the likelihood of also t being the nearest neighbour of x. Only when both these likelihoods are high will MP also

[1] `https://fm4.orf.at/soundpark`

[2] Currently not accessible due to Adobe Flash Player's phaseout beginning of 2021.

achieve a high value. To estimate MP, we follow the approach proposed by [11] and use the empirical distribution to compute

$$\mathrm{MP}(d_{x,t}) = \frac{|\ \{j : d_{x,j} > d_{x,t}\} \cap \{j : d_{t,j} > d_{t,x}\}\ |}{n}. \quad (2)$$

Here $d_{x,t}$ is short for $D_{\mathrm{SKL}}(\mathcal{G}_x, \mathcal{G}_t)$ and corresponds to the symmetric KL divergence between the Gaussians of x and t; n is the total number of objects. We turn MP into a distance by defining $D_{\mathrm{MP}}(d_{x,t}) = 1 - \mathrm{MP}(d_{x,t})$.

4.4 The Attack

In [7], an attack originally proposed by Carlini and Wagner [2] for speech processing was used to compute adversarial examples for the real-world recommender system described above. The Carlini & Wagner (C&W) attack is a targeted white-box adversarial attack, meaning it requires full knowledge of a system under attack and aims at changing the output of the system to a predefined target t. In our case, the objects to be adversarially modified are audio files representing Soundpark songs, and the *adversarial perturbations* δ we wish to apply to these are modifications to the audio, in the form of additive noise, that are (hopefully) imperceptible. To achieve a desired distorted system prediction, which in our case is an increased number of times a particular song is recommended, we minimise a system loss with respect to the target output iteratively. In this modification of the attack, the target corresponds to a song within the dataset that is already a hub, and hence is often recommended. We simultaneously try to keep the added adversarial perturbation δ as small (or imperceptible) as possible by also minimising the squared L2-norm of a perturbation.

In order to compute an adversarial perturbation δ for a particular song, with the aim of increasing its recommendations, we therefore try to minimise a combination of a system loss and the norm of the perturbation iteratively. This results in an optimisation objective and an update formula for δ as follows, which we can realise with gradient descent:

$$L_{\mathrm{total}} = \|\delta_{ep}\|_2^2 + \alpha * D_{\mathrm{SKL}}(\mathcal{G}_{x+\delta_{ep}}, \mathcal{G}_t) \quad (3)$$

$$\delta_{ep+1} = \mathrm{clip}_\epsilon(\delta_{ep} - \eta * \mathrm{sign}(\nabla_{\delta_{ep}} L_{\mathrm{total}})). \quad (4)$$

Note that a perturbation δ is here subscripted with the current epoch ep, and its updates are performed based on the sign of the gradient $\nabla_{\delta_{ep}}$ (w.r.t. δ_{ep}) and the factor η. Updates are further clipped in each iteration to stay between $-\epsilon$ and ϵ. The system loss of the music recommender here is represented by the KL divergence (D_{SKL}), and $\mathcal{G}_x$ represents a Gaussian of a particular song x. The multiplicative factor α balances the focus of the optimisation between finding perturbations more quickly or less perceptible. For each x, the target song t is chosen to be its closest hub song. A perturbation δ is updated until the attack is successful (stopping criterion), i.e., song x has a $O^k \geq 25$, or until 500 update steps were made. The perturbation δ_0 is initialised with zeros.

After a successful attack, the number of times a song is recommended by the system is therefore higher than before. This could be exploited in systems like the recommender of the FM4 Soundpark, as users can directly contribute to the song catalogue and could try to submit a perturbed version of their song in order to manipulate the recommender.

4.4.1 *Modified Mutual Proximity Attack*

The attack described above uses knowledge about the song encoding and distance measure used by the recommender (via $\mathcal{G}$ and D_{SKL}), but not about the specific defence (MP) it will face. In order to create a more difficult defence scenario for MP, we additionally incorporate knowledge of the defence into a modified adversarial attack.

More precisely, we attempt to attack the system by (I) leaving the main objective L_{total} unchanged (cf. Equation (3)) and only adapting the stopping criterion of the attack. The stopping criterion decides if an attack is successful, which is now only the case if the k-occurrence is at least 25 *after* distances between all objects in a dataset are rescaled with MP. This is different to the original case, in which the attack was successful if the k-occurrence exceeded 25 *without* applying MP. In what follows, we call this attack adaptation *C&W*$_{KL}^{mod}$. For the second adaptation (II), the stopping criterion is adapted in the same way; additionally, we change the objective L_{total} such that we minimise an approximation of the MP between a song x and its target t (cf. Equation (2)) instead of the KL divergence. This approximation is necessary as MP itself is not differentiable. Including this knowledge results in an updated L_{total} in Equation (3) of

$$L'_{\mathrm{total}} = \|\delta_{ep}\|_2^2 + \alpha * \widetilde{D}_{\mathrm{MP}}(d_{x+\delta_{ep},t}), \quad (5)$$

with

$$\widetilde{D}_{\mathrm{MP}}(d_{x,t}) = 1 - \frac{\sum_i \mathrm{bt}_{i,t}(x) * \mathrm{bt}_{i,x}(t)}{n}, \quad (6)$$

$$\mathrm{bt}_{i,t}(x) = \max(\tanh(d_{x,i} - d_{x,t}), 0). \quad (7)$$

Here $\widetilde{D}_{\mathrm{MP}}$ corresponds to an approximation of $D_{\mathrm{MP}}(d_{x,t})$, i.e., $\widetilde{D}_{\mathrm{MP}} \approx 1 - \mathrm{MP}(d_{x,t})$. The numerator in Equation (6) is the approximation of the numerator in Equation (2), and consists of summation over i, where i denotes all elements in our data catalogue. The constant n in the denominator denotes the total number of objects in the catalogue. The function bt is a differentiable approximation of the *bigger-than* function in Equation (2); here the function $\tanh$ denotes the hyperbolic tangent function, and $\max$ returns the maximum of its two inputs. Subsequently, this modified attack will be denoted by *C&W*$_{MP}^{mod}$.

5. EXPERIMENTS

To allow reproducibility of the experiments summarised in this work, the code as well as all necessary attack-parameters are available on Github [3] .

[3] https://github.com/CPJKU/hub_defence

5.1 Parameters of Attacks

The adversarial attack proposed in [7] and our modifications require a set of parameters, which influence the number of successful adversarial examples as well as their quality. More precisely, the parameters we need to choose are the clipping factor ϵ, the multiplicative factor η which determines the step-size of the gradient updates, and finally factor α, which is responsible for controlling the focus of the attack on finding a higher number versus less perceptible adversaries. For the original attack, we tried to reproduce the results shown in [7] and used the proposed parameters, i.e. $\epsilon = 0.1, \eta = 0.001, \alpha = 25$.

For the remaining attacks, we performed a grid-search over various parameter combinations, and chose the settings in which we found the overall highest number of successful adversarial examples. For C&W$_{KL}^{mod}$, this corresponds to $\epsilon = 1.0, \eta = 0.001, \alpha = 25$; For C&W$_{MP}^{mod}$, we set $\epsilon = 1.0, \eta = 0.0005$, optionally with $\alpha = 100$. Note that this grid-search was done on the complete data base, since the white-box nature of the attacks requires full knowledge of the attacked system (here: all Gaussians). Additionally an attacker would naturally also have knowledge of the audio to be perturbed, hence a distinction into train and test data for parameter estimation, as is customary in most machine learning settings, is not necessary when evaluating such white-box attacks.

5.2 Mutual Proximity as a Defence

Before using MP as a defence, we investigate the vulnerability of the undefended recommender by applying the C&W attack used in related work [7]. The first line in Table 5.2 shows the result of this attack.

The columns in Table 5.2 depict the nature of the attack, the number of initial hubs (i.e., before the attack), the number of adversarial hubs (i.e., songs for which the attack was successful), and the number of songs for which the attack was not successful (# Non-hubs). The last two columns contain the average $\pm$ standard deviation of the Signal-to-Noise ratio (SNR) and the k-occurrence of successful adversarial examples. We manage to successfully attack around 44.1% of all files with an average SNR of roughly 39.0dB, which is similar to the results shown in [7]. Note that we use a subset of the data used in [7].

Next, we use MP to defend against future adversarial attacks by integrating it in the music recommender. We first rescale the distances between all original (clean) objects in our dataset with MP. Beyond what is shown in Table 5.2, let us briefly look at some additional information displaying the impact of applying MP on the data. Before rescaling our dataset has 1,663 (33.3%) anti-hub and 202 (4.0%) hub songs, with a maximal k-occurrence of $O^k = 393$. This means that one third of the song collection is never recommended without MP. After rescaling however, this is reduced to 408 (8.2%) anti-hubs and only 3 (0.1%) hubs, with a maximal k-occurrence of $O^k = 35$. To further advocate the usage of MP, we can also look at the average retrieval accuracy R^k as defined in [11], which increases from 43.7% before MP to 47.7% after MP.

After applying MP, we use the two adaptations of the attack in an attempt to find adversarial hubs for the defended system. Going back to Table 5.2, the second line shows that the success rate of the attack decreases from 44.1% to only 2.5% when we now try to minimise the KL divergence between a song and its target (C&W$_{KL}^{mod}$). Also the SNR of the successful adversaries decreases from on average 39.0dB to 28.3dB. If we use the second proposed adaptation (C&W$_{MP}^{mod}$), and minimise the MP instead, the SNR is higher (42.9dB), however we are only successful for a small percentage (0.4%) of all files (see third line). In an attempt to find a larger number of successful adversaries to test the boundaries of MP as a defence, we repeat the attack using the adaptation C&W$_{MP}^{mod}$ once more, yet leaving out the factor $\|\delta_{ep}\|_2^2$ restricting the norm of the perturbation (i.e., minimising $L'_{\text{total}} = \widetilde{D}_{\text{MP}}(d_{x+\delta_{ep},t})$). The result is shown in the last line of Table 5.2. Disregarding the perceptibility of a perturbation during the optimisation process leads to a slightly higher success-rate of 5.7%, but also in a low average SNR of 20.4dB. The perceptual differences of the different attacks can be examined in the supplementary material [4], where listening examples are provided. The song excerpts are chosen to represent good as well as bad examples for particular attacks in terms of their perceptibility. Note that perturbations with a SNR of above $40dB$ are hardly perceptible; SNRs between $40-20dB$ are perceptible, at least when compared to the original audio, but without noticeably changing the essence of the song. SNRs lower than $20dB$ tend to become clearly perceptible or even disruptive.

5.3 Post-Hoc Defence

Instead of integrating MP into the system directly, MP could also be applied post-hoc to rescale the distances between all files *after* an attack , and hence post-process its recommendations. While in a real-world scenario it should be preferable to directly integrate the defence into a system to make an attack as difficult as possible, we want to briefly show that MP could also reduce the impact of an adversarial attack on an undefended system. For each of the $2,206$ hub songs we found when attacking the undefended system, we apply MP after one such song is added to the (otherwise clean) dataset, and observe how its k-occurrence and therefore hubness is changed due to the transformation.

For $2,193$ out of the $2,206$ (i.e., 99.4%) files we manage to decrease the k-occurrence enough to revert them back to *normal* songs (i.e., $0 < O^k < 25$). In other words, MP could also be a successful post-hoc defence for the C&W attack on the music recommender.

6. DISCUSSION

In this work we examined a specific real-world system to investigate the suitability of MP as an adversarial defence, which is why we briefly want to reason about how our findings could generalise to other settings. A limiting requirement of this defence is, due to its nature, that the attack is

[4] https://cpjku.github.io/hub_defence

Adaptation	# Initial Hubs	# Adversarial Hubs	# Non-hubs	SNR	O^k
original	202 (4.0%)	2,206 (44.1%)	2,592 (51.8%)	39.0 ± 5.1	41.7 ± 23.0
C&W$_{KL}^{mod}$	3 (0.1%)	126 (2.5%)	4,871 (97.4%)	28.3 ± 6.6	25.9 ± 1.3
C&W$_{MP}^{mod}$	3 (0.1%)	18 (0.4%)	4,979 (99.6%)	42.9 ± 7.8	26.5 ± 1.5
C&W$_{MP}^{mod}$ (no norm)	3 (0.1%)	286 (5.7%)	4,711 (94.2%)	20.4 ± 9.2	26.1 ± 1.4

Table 1. Results of the (adapted) C&W attacks. The columns are the adaptation of the attack, the number of initial and adversarial hubs, the number of non-hubs after the attack, and the average SNR and k-occurrence of successful adversaries.

aimed at exploiting the *hubness* issue in some way. However, we expect the proposed defence not to depend on the exact computation of adversarial examples, and hence to be useful against approaches different from the C&W-like attack.

As previously mentioned, hubness has also been shown to be an issue in various different (recommender) systems, most importantly in systems based on collaborative filtering [13, 14]. In other words, also systems like these are susceptible to attacks exploiting hubness, and could potentially be suited for and benefit from the proposed adversarial defence. Note here once more that hubness in a recommender system can lead to unfair recommendations, as a significant part of data is never recommended, while a relatively small part of the data is recommended very often (cf. [9]); adversarial attacks that exploit hubness additionally amplify this effect on the fairness of recommendations, which is why a defence against them is crucial. However, as MP was shown to be able to reduce the hubness for various kinds of data (cf. [16]), we assume that the suitability of MP as a defence extends to a variety of different systems based on diverse data.

A further point for discussion is the potentially high computational complexity of the methods applied in this work. While the original definition of MP has quadratic complexity (w.r.t. the dataset size n), approximations that are linear in n were previously proposed to allow an application of this method even to large datasets (cf. [11, 17]). The high complexity also carries over to the modified adversarial attack, in which we therefore also used an approximation of MP. Nevertheless, the computational cost increases with growing dataset sizes, which is problematic in particular for the informed (modified) attack, as the MP needs to be computed in every iteration. The defence, however, requires additional computations of MP only if new data points are added to a catalogue.

Our work also connects to recent results establishing that, as the local intrinsic dimensionality of data increases, nearest neighbour classifiers become more vulnerable to adversaries [21]. Datasets are often embedded in spaces of higher dimensionality than is needed to capture all their information. The minimum number of features necessary to encode this information is called intrinsic dimensionality (see [22] for a recent review). It is well known that hubness also depends on a dataset's intrinsic dimension [10]. Future work should explore this possible new link of the concepts of adversaries, high dimensionality and hubness.

7. CONCLUSION

In this work, we have evaluated an existing hubness reduction method (Mutual Proximity) as a defence against an adversarial attack on a real-world music recommender. Before the defence, the attack is able to artificially inflate playcounts of songs at the expense of other songs never being played, resulting in a very unfair recommender. While the success rate of the attack is around 44.1% before the defence with MP, we decrease it to $0.4\% - 5.7\%$ after the defence, despite incorporating knowledge of the defence in the attacks. In addition to making it more difficult to find successful adversarial perturbations, the defence also forces the attack to result in more perceptible perturbations. As we defend a specific real-world recommender in this work, we also discussed how this could generalise to other systems, and reason that it could be a suitable defence against diverse kinds of hubness-related attacks in the future.

Acknowledgments

This research was supported by the Austrian Science Fund (FWF, P 31988). GW's work is supported by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation programme, grant agreement No 101019375 (*Whither Music?*). For the purpose of open access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

8. REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. of the 2nd International Conference on Learning Representations, ICLR*, 2014.

[2] N. Carlini and D. A. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. of the 2018 IEEE Security and Privacy Workshops, SP Workshops*. IEEE, 2018, pp. 1–7.

[3] B. L. Sturm, "A simple method to determine if a music information retrieval system is a "horse"," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1636–1644, 2014.

[4] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proc. of the 25th Annual Network and Distributed*

System Security Symposium, NDSS. The Internet Society, 2018.

[5] D. Jakubovitz and R. Giryes, "Improving DNN robustness to adversarial attacks using jacobian regularization," in *Proc. of the 15th European Conference on Computer Vision, ECCV*, ser. Lecture Notes in Computer Science, vol. 11216. Springer, 2018, pp. 525–541.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. of the 6th International Conference on Learning Representations, ICLR*, 2018.

[7] K. Prinz, A. Flexer, and G. Widmer, "On end-to-end white-box adversarial attacks in music information retrieval," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 93—-104, 2021.

[8] M. Gasser and A. Flexer, "Fm4 soundpark: Audio-based music recommendation in everyday use," in *Proc. of the 6th Sound and Music Computing Conference, SMC*, 2009, pp. 23–25.

[9] A. Flexer and J. Stevens, "Mutual proximity graphs for improved reachability in music recommendation," *Journal of New Music Research*, vol. 47, no. 1, pp. 17–28, 2018.

[10] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *Journal of Machine Learning Research*, vol. 11, no. 86, pp. 2487–2531, 2010.

[11] D. Schnitzer, A. Flexer, M. Schedl, and G. Widmer, "Local and global scaling reduce hubs in space," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2871–2902, 2012.

[12] F. Pachet and J.-J. Aucouturier, "Improving timbre similarity: How high is the sky," *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, pp. 1–13, 2004.

[13] P. Knees, D. Schnitzer, and A. Flexer, "Improving neighborhood-based collaborative filtering by reducing hubness," in *Proc. of the 4th International Conference on Multimedia Retrieval, ICMR*. ACM, 2014, p. 161.

[14] K. Hara, I. Suzuki, K. Kobayashi, and K. Fukumizu, "Reducing hubness: A cause of vulnerability in recommender systems," in *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 815–818.

[15] D. Schnitzer, A. Flexer, and N. Tomasev, "A case for hubness removal in high-dimensional multimedia retrieval," in *Proc. of the 36th European Conference on Information Retrieval Research, ECIR*, ser. Lecture Notes in Computer Science, vol. 8416. Springer, 2014, pp. 687–692.

[16] R. Feldbauer and A. Flexer, "A comprehensive empirical comparison of hubness reduction in high-dimensional spaces," *Knowledge and Information Systems*, vol. 59, no. 1, pp. 137–166, 2019.

[17] R. Feldbauer, M. Leodolter, C. Plant, and A. Flexer, "Fast approximate hubness reduction for large high-dimensional data," in *Proc. of the 2018 IEEE International Conference on Big Knowledge, ICBK*. IEEE Computer Society, 2018, pp. 358–367.

[18] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

[19] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. P. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019, pp. 3353–3364.

[20] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Accelerating adversarial training via maximal principle," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019, pp. 227–238.

[21] L. Amsaleg, J. Bailey, D. Barbe, S. Erfani, M. E. Houle, V. Nguyen, and M. Radovanović, "The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality," in *Proc. of the 2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE, 2017, pp. 1–6.

[22] F. Camastra and A. Staiano, "Intrinsic dimension estimation: Advances and open problems," *Information Sciences*, vol. 328, pp. 26–41, 2016.