Assessing GAN-based approaches for generative modeling of crime text reports

Samira Khorshidi and George Mohler
Computer and Information Science
Indiana University - Purdue University Indianapolis
Indianapolis, USA
{sakhors,gmohler}@iupui.edu

Jeremy G. Carter
O'Neil School of Public and Environmental Affairs
Indiana University - Purdue University Indianapolis
Indianapolis, USA
carterjg@iupui.edu

***Abstract*—Analysis and modeling of crime text report data has important applications, including refinement of crime classifications, clustering of documents, and feature extraction for spatio-temporal forecasts. Having better neural network representations of crime text data may facilitate all of these tasks. This paper evaluates the ability of generative adversarial network models to represent crime text data and generate realistic crime reports. We compare four state of the art GAN algorithms in terms of quantitative metrics such as coherence, embedding similarity, negative log-likelihood, and qualitatively based on inspection of generated text. We discuss current challenges with crime text representation and directions for future research.**

***Index Terms*—Crime Reports, GAN, Coherence, Text representation**

I. Introduction

Crime events reported to law enforcement consist of event date and time, spatial location, and also may contain a short narrative description of the event. Recently, crime topic models [1] have been introduced for crime report narratives to better classify events into more nuanced and accurate categories. These models have better coherence than human-defined categories and may lead to better spatial risk estimates [2]. Crime event text data also arises in social media, for example, Twitter [3] and Nextdoor [4], and machine learning methods have been developed to both detect such events, classify them, and use them in spatio-temporal forecasts. More recently, neural network based representations in the form of restricted Boltzmann machines (RBM) have been introduced for crime report data [5]. These methods can be used for event clustering and crime linkage [6] (identifying crimes committed by the same individual).

Generative adversarial networks (GANs) [7] are a recently introduced framework for deep generative modeling of high-dimensional, complex data. GANs consist of two networks: one network that maps from a latent space random Gaussian vectors to synthetic observations (e.g., synthetic images or text reports), and a second network that learns to distinguish between real and synthetic data. Upon successful (adversarial) training, the GAN has learned an accurate representation of the data, and synthetic and real examples may be indistinguishable. Thus GANs have an advantage over other unsupervised deep learning approaches such as RBMs in that qualitative evaluation of the representation is possible through inspection of generated images or text. While much of the advances in GAN research are with respect to image data, several methods have recently been introduced for generative modeling of text. Our goal in this paper is to evaluate these methods for the purpose of generative modeling of crime text data.

The outline of the paper is as follows. In Section II, we review machine learning techniques used in criminology, generative adversarial networks, and the application of GANs to text data. In Section III, we provide details on the four GAN approaches we analyze in this paper. In Section IV, we introduce the dataset we use for the evaluation, consisting of text narratives from Los Angeles crime reports. In Section V, we describe the evaluation metrics used in our study, and we present the results of the evaluation in Section VI. We discuss our results and directions for future work in Section VII.

II. Related work

In this section, we review related work on machine learning in criminology and generative adversarial networks.

A. Machine learning in criminology

A number of studies have focused on applying machine learning methods for space-time crime forecasting. Point process models are used to model space-time clustering [8], [9] and gun violence patterns on social networks [10]. More recently, deep learning [11], [12], and learning to rank [13], [14] approaches have been applied to space-time crime forecasting, showing improved accuracy. Deep learning models have also been used to infer crime rates from Google street view images [15]. Auxiliary data has been shown to improve space-time models, including location data [16] and geolocated Twitter data [3].

Other recent work in machine learning and crime has focused on text data. Wang et al. [3] extract topics from Twitter posts through latent Dirichlet allocation (LDA) and then use a general linear model to predict Hit And Run crimes. Kuang et al. [1] consider latent behavioral and situational conditions underlying crimes and then use latent distributions through topic modeling to cluster and classify crime categories. Al-Zaidy et al. [17] proposed an automated framework to extract information from unstructured text data, including e-mails, chat logs, blogs, web-pages, and text documents, to discover criminal networks. More recently, partially Generative Neural Networks (PGNNs) [18] have been used to classify gang crimes based on text reports. However, in that study, the quality of generated crime text data was not evaluated.

B. Generative adversarial networks

Generative adversarial networks (GANs) are a class of unsupervised deep learning models [7], [19]. Each GAN has two primary components – a Generator Neural Network (G) and Discriminator Neural Network (D). The generator $G(z)$ takes an input, z, a sample from probability distribution $p(z)$ (typically Gaussian), and generates synthetic (fake) data. The discriminator takes as input either a real observation, in our case crime report, or synthetic output from the generator, and determines through binary classification whether the input is real or fake. The two networks are trained jointly, where alternating gradient descent steps are made to i) train the generator by freezing the weights of the discriminator and then update the weights of the generator so as to increase the probability that fake data is labeled real and ii) train the discriminator to correctly classify real and fake data. Upon successful training, the two networks reach an equilibrium in what can be viewed as a minimax game [20]. Much of the success of GANs has occurred for continuous data in the context of images [21] [22] [23] [24] [25], and using GAN on discrete data such as text is a challenging task.

C. Generative adversarial networks as a language model for text generation

Using GANs for generative modeling of discrete data is challenging [7], and GANs often do not have consistency in long-term syntactic generation [26]. To overcome these challenges, Zhang et al. [27] use new architectures for each GAN network, in particular, Long short-term memory (LSTM) cells as the generator and convolutional neural networks (CNNs) for the discriminator. Then, using a confusion training strategy, the discriminator learns the sentence structure. Instead of data distribution matching, Zhang et al. employ feature distribution matching. In this case, the training objective that reflects feature mapping is as follows:

$$\begin{aligned} \min L_D =& - E_{s\sim S} \log D(s) - E_{z\sim p_z(z)} \log[1 - D(G(z))] \\ \min L_G =& tr(\Sigma_s^{-1}\Sigma_r + \Sigma_r^{-1}\Sigma_s) + \\ & (\mu_s - \mu_r)^T(\Sigma_s^{-1} + \Sigma_r^{-1})(\mu_s - \mu_r) \end{aligned} \tag{1}$$

Here Σ_s and Σ_r are the covariances of the real and fake feature vectors, and μ_s and μ_r are their means.

Another GAN based model for text generation is TextGAN [28], which adds two terms to the regular objective function of GAN: i) Euclidean distance between the reconstructed latent code and the original code, and ii) Maximum Mean Discrepancy (MMD) (i.e., a distance on the space of probability measures [29]). Consequently, the discriminator has to produce the most discriminative, representative, and challenging sentence features. The generator then has to match these features. The objective function of TextGAN is given by:

$$\begin{aligned} &\max_D L_D = L_{GAN} - \lambda_r L_{recon} + \lambda_m L_{MMD^2} \\ &\min_G L_G = L_{MMD^2} \\ &L_{GAN} = E_{s\sim S} \log D(s) - E_{z\sim p_z(z)} \log[1 - D(G(z))] \\ &L_{recon} = ||\hat{z} - z|| \end{aligned} \tag{2}$$

Here $\hat{z}$ is the reconstructed latent code, z is the original code drawn from the prior distribution $p_z(\cdot)$, and L_{MMD^2} is the Maximum Mean Discrepancy (MMD). Ultimately, Zhang et al. [28] conclude recurrent neural networks (RNN) in text generation result in exposure bias since, at each step, the generated word is based on previous ones. This error is proportional to the sentence's length and prone to occur in the general GANs. Lamb et al. [30] solves this problem by proposing "professor forcing architecture", using an extra discriminator trained to maintain stable long term dependency within words in a sentence.

III. An evaluation of four GAN models for crime report text generation

In this research, we compare four GAN methods for crime report text generation: SeqGAN [24], MaliGAN [26], LeakGAN [31] and RankGAN [32]. In Table I, we provide an overview of the characteristics of these four GAN models. The first approach is SeqGAN, which addresses exposure bias in maximum likelihood inference and the challenges of generating discrete data with GANs. The generator is a Recurrent Neural Network (RNN) with Long short-term memory (LSTM) cells to deal with vanishing gradients, and the discriminator is a CNN. Reinforcement learning (RL) is used for model training, where generated tokens are states, and actions are the next token to be generated. In this framework, the discriminator releases rewards based on a complete sequence of tokens.

TABLE I
Comparison of utilized models

Model	Discriminator	Generator	Loss	Training
SeqGAN	CNN	RNN	MLE	Policy gradient
MaliGAN	CNN	RNN	D: MLE G: MLE importance sampling	Policy gradient
RankGAN	CNN ranker regressor	LSTM	D: Ranking	Relative ranking information Policy gradient
LeakGAN	CNN	Hierarchical LSTM	MLE	Interleaved Training

The second method is MaliGAN, which is based on an alternative objective function. The idea behind MaliGAN is to use the information of the discriminator as an additional training signal. The generator is trained by importance sampling, whereas the discriminator objective function is the same as the original GAN [7]. RankGAN is the third model we evaluate, where the learning process is defined by relative ranking information between the machine-written and the human-written sentences in an adversarial context. The discriminator is a CNN rank regressor, and the training of the generator (LSTM network) is by the policy gradient technique. Within the context of policy gradients, the ranking scores of the ranker network are viewed as rewards to learn for the language generator [32]. For the ranking score of a generated sentence, a softmax operation is applied over relevance scores of the generated sentence against all sentences in a comparison set. Following the min-max game of the original GAN [7], the Ranker (R) tries to rank the generated data lower than the real data, and the generator is trying to generate more realistic documents.

LeakGAN, the fourth model we consider, attempts to address semantic loss over long sentences observed in SeqGAN, MaliGAN, and RankGAN. LeakGAN deviates from binary classification when designing the discriminator. In addition to such a scalar reward, the discriminator passes extracted high-level features to the generator within a reinforcement learning framework. The generator has a hierarchical structure consisting of Manager and Worker networks. The Manager network (LSTM) takes the extracted high-level feature vector coming from the discriminator and passes the latent vector to the Worker network. The Worker network then generates the next word based on the latent vector of the current word. Hence LeakGAN addresses both the non-informativeness and the sparsity issues of the rewards. Interleaved training, which is a combination of supervised learning (MLE) and adversarial learning (GAN), is used to find the parameters of LeakGAN.

IV. Data

The dataset that we have employed in this study consists of text narratives from crime reports provided by the Los Angeles Police Department (LAPD) from 1/1/2009 to 7/19/2014. The data includes 805556 crime reports in total. We give an example of a real crime report text narrative in Table III.

Preprocessing consisted of restoring incomplete words in the data presented in Table II. To train GAN models, we generate tokens of our narratives using the ‘nltk‘ software package [33], and our vocabulary size is 5000 words.

V. Evaluation

To evaluate the four GAN models, we use three metrics: 1) negative log likelihood, 2) embedding similarity, and 3) coherence. Negative log-likelihood is the average negative log-likelihood of each GAN model evaluated on the real test data. Embedding similarity is related to BLEU [34], which measures

TABLE II
INCOMPLETE WORDS AND THEIR RESTORED VERSION

Word	restored version	Word	restored version
vic	victim	victs	victims
v1	victim	v2	victim
v	victim	neg	negative
s	suspect	ss	suspect
sus	suspect	susps	suspects
susp	suspect	s1	suspect
remvd	removed	rmvd	removed
att	attack	stillinside	still inside
cointinuously	continuously	lemmon	lemon
ent	enter	donttell	dont tell
loc	location	dir	direction
resid	resident	veerbal	verbal
veh	vehicle	thransaction	transaction
prop	property	usedcredit	used credit
unk	unknown	beerbottle	beer bottle

TABLE III
EXAMPLE CRIME REPORT TEXT NARRATIVE.

s approached v and asked v to fight s hit v multiple times v defended himself and punched s. s fled location southbound on kingsley from melrose.

whether n-grams generated by the model appear in a held-out corpus. Inspired by BLEU, embedding similarity (EmbSim) is a way to estimate text-similarity. Instead of matching sentences words by words, EmbSim compares their word embeddings. For each word embedding for real text data, the cosine distance with other words is computed. Then a matrix W is constructed, where $w_{ij} = cos(e_i, e_j)$ and e_i is the word embedding of the ith word. We call W the similarity matrix of real data. Likewise, the word embedding of generated fake data will be evaluated, and we get the similarity matrix W’ of a generated crime text report. Here $w'_{ij} = cos(e'_i, e'_j)$, e'_i is the word embedding of the word i in the generated data. The EmbSim is defined as follows:

$$EmbSim = \log(\Sigma_{i=1}^{N} \frac{cos(W'_i, W_i)}{N}) \tag{3}$$

Where W_i is the i^{th} column of W [35].

The third metric we consider is a sentence coherence score. First, we apply topic modeling to real and generated narratives and extract the latent topics. Topic modeling can be seen as an adaptive classification that is adaptive to emerging crime categories without human supervision. We apply both Latent Dirichlet Allocation (LDA) and Mallet’s LDA to extract keywords for each latent topic in the data. Mallet’s LDA extracted topics and keywords in our real data have been provided in Table IV. Then we compare the coherence score C_v based on a sliding window, a one-set segmentation of the top words, and an indirect confirmation measure using normalized point-wise mutual information (NPMI) and the cosinus similarity [36].

As shown in Fig. 1, topics in our real data are overlapping. A good topic model has big and non-overlapping bubbles scattered throughout the chart instead of being clustered in

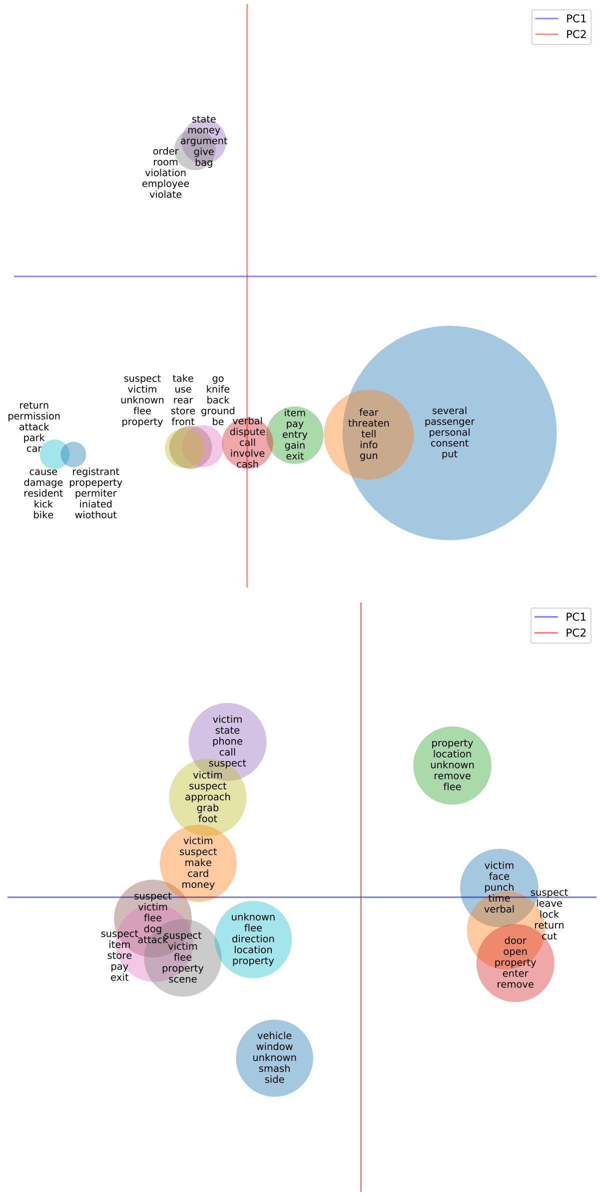

Fig. 1. Intertopic Distance Map. We have reduced dimension via Jensen Shannon Divergence and Principal Coordinate Analysis(PCoA). Top: LDA, Down: MalletLDA. Each bubble represents a topic. The larger the bubble, the more prevalent is that topic.

one quadrant. However, overlapping topics is normal when a wide set of intuitively related topics exist in the data.

The sentence coherence score of a narrative entry determines to what extent the topic of a generated narrative is consistent, for example, if it is clear the narrative is a robbery, or if the generated sentence contains inconsistent topics (for example, words associated with burglary mixed in with words associated with homicide). In our proposed sentence coherence score, Equation 4, we estimate the connection of a narrative to available topics in our real data. A lower score means the sentence contains more topics; a higher score means the sentence a lower number of topics.

TABLE IV
KEYWORDS IN EACH TOPICS FOR MALLET'S LDA.

Topic ID	Keywords
0	victim, face, punch, time, verbal, strike, push, argument, dispute, hit
1	suspect, leave, lock, return, cut, bike, park, victim, secure, room
2	property, location, unknown, remove, flee, enter, victim, unsecured, unlocked
3	door, open, property, enter, remove, unknown, gain, entry, front, rear
4	victim, state, phone, call, suspect, kill, demand, threaten, order, fear
5	suspect, victim, flee, dog, attack, approach, screwdriver, vand, tie, res
6	suspect, item, store, pay, exit, location, business, walk, entered, purse
7	suspect, victim, flee, property, scene, removed, location, porch, package, unknown
8	victim, suspect, approach, grab, foot, approached, drive, fire, pull, hand
9	unknown, flee, direction, location, property, tool, remove, tire, type, slash
10	vehicle, window, unknown, smash, side, break, damage, hard, driver, pass
11	victim, suspect, make, card, money, permission, check, info, credit, personal

$$Coherence_{narrative} = [\frac{count(W_u) - count(W_c)}{count(W_t)}]^+ \quad (4)$$

Here $[\cdot]^+ = max(0, \cdot)$. W_u is a word present in less than two topics, W_c is any topic keyword and, W_t is a any word in the narrative.

VI. RESULTS

In this section, we compare SeqGAN, LeakGAN, MaliGAN, and RankGAN applied to the LAPD crime report narratives in terms of the three evaluation metrics described above. Negative log-likelihood and embedding similarity results are reported in Table V. Based on these results, LeakGAN has the lowest (best) NLL and a relatively high (better) embedding similarity. As mentioned, we have employed Latent Dirichlet Allocation (LDA) and Mallet's LDA to extract latent topics and their keywords to construct the sentence coherence score. The coherence scores are presented in Table VI, showing that SeqGAN has the best Mallet's LDA-coherence and LeakGAN has the best LDA-coherence.

We also qualitatively inspect the generated test and present example generated narratives for each model in Table VII. In the first column of the Table VII, we have reported our sentence coherence score. We see realistic generated phrases such as "suspect approach victim vehicle cause damage," though we also see some examples that lack coherence and in some cases, contain misspelled words. By examining each model's generated narratives, we observe that RankGAN and LeakGAN favor generating longer narratives in contrast to SeqGAN and MaliGAN. However, this does not necessarily lead to more realistic or meaningful narratives.

TABLE V

METHOD APPLICATION COMPARISON USING NEGATIVE LOG LIKELIHOOD AND EMBEDDING SIMILARITY

Method	NLL(epoch 180)	EmbedSim(epoch 180)
SeqGAN	$1.63E+00$	$-1.13E-02$
MaliGAN	$2.47E+00$	$-9.96E-3$
LeakGAN	$1.25E+00$	$-1.21E-02$
RankGAN	$2.85E+00$	$-9.84E-03$

TABLE VI

COHERENCE SCORE OF LDA AND MALLET'S LDA ON REAL DATA. HIGHER COHERENCE SCORE MEANS BETTER PERFORMANCE.

Data source	Model	C_v Coherence score [36]
Real data	LDA	$2.88E-01$
	Mallet's LDA	$5.72E-01$
SeqGAN generated	LDA	$4.62E-01$
	Mallet's LDA	$4.23E-01$
MaliGAN generated	LDA	$3.96E-01$
	Mallet's LDA	$3.72E-01$
LeakGAN generated	LDA	$4.75E-01$
	Mallet's LDA	$4.11E-01$
RankGAN generated	LDA	$4.26E-01$
	Mallet's LDA	$3.91E-01$

TABLE VII

METHODS AND TWO GENERATED NARRATIVES BY EACH METHOD. SCORE IN THIS TABLE REFERS TO OUR SENTENCE COHERENCE SCORE COMPUTED BASED ON EXTRACTED TOPICS BY MALLET'S LDA MODEL

Score	Narrative sample
	Real
0.66	suspect continiously calls and text victim harrassing messages victim has repeatedly advised suspect to stop making contact
0.0	suspect approached victims vehicle punctured victims tires with an unknown sharp tool and fled location in unknown direction
	SeqGAN
0.38	a unknown suspect rmvs rent vehicle roll passenger door damage victim vehicle around vehicle cause deep dent
0.0	unknown suspect rmvs smash victims vehicle windows passenger windows vehicle
	MaliGAN
0.85	elm middle of former friends apt victim made unauthorized charge accounts in the secured her namewithout victim if he could not give anyone anyone to defraudbank
0.2	elm mutual partner which occured suspect brother or care of victims suspect fled with victim property wo knowledge
	LeakGAN
0.0	suspect approach victims vehicle cause damage unknown suspect usedhis kindle carbon merchandis
0.33	suspect approach victim behind punch victim face victim andforcibly psych opcredit windows1 flee location
	RankGAN
0.0	unknown suspect t6ook victims property got to vehicle
0.5	unknown suspect t6ook victims vehicle victim was brockton radiored stevely occs thnfled cummings mrcedes to unknown suspects akas v006 s1app forgets 1145 hrs and 242/243pc 30000usd merchandise failed in a automotivegarage

VII. DISCUSSION AND CONCLUSION

In this research, we reviewed the recent methodology for GAN based text generation and applied four such methods to the task of generating crime narratives. We found that LeakGAN performed the best in terms of negative log-likelihood and embedding similarity, whereas LeakGAN and SeqGAN had the best (and comparable) LDA based coherence. Future work may focus on improving GAN based narrative generation, as our qualitative analysis shows that GAN generated crime reports are not quite at the level of human quality. Another line of future research would be to use GAN based embeddings for specific applications. For example, Bi-direction GANs [37] may be used to invert the generator network and embed crime reports in the latent space. Such embeddings may be useful for crime linkage analysis, anomaly detection, and crime clustering and categorization.

VIII. ACKNOWLEDGEMENTS

This research was supported by NSF grants SCC-1737585 and ATD-1737996.

REFERENCES

[1] D. Kuang, P. J. Brantingham, and A. L. Bertozzi, "Crime topic modeling," *Crime Science*, vol. 6, no. 1, Dec 2017.

[2] R. Pandey and G. O. Mohler, "Evaluation of crime topic models: topic coherence vs spatial crime concentration," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2018, pp. 76–78.

[3] X. Wang, M. S. Gerber, and D. E. Brown, "Automatic crime prediction using events extracted from twitter posts," in *Social Computing, Behavioral - Cultural Modeling and Prediction*, S. J. Yang, A. M. Greenberg, and M. Endsley, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 231–238.

[4] G. Mohler and P. J. Brantingham, "Privacy preserving, crowd sourced crime hawkes processes," in *2018 International Workshop on Social Sensing (SocialSens)*. IEEE, 2018, pp. 14–19.

[5] S. Zhu and Y. Xie, "Crime event embedding with unsupervised feature selection," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3922–3926.

[6] ——, "Spatial-temporal-textual point processes with applications in crime linkage detection," *arXiv preprint arXiv:1902.00440*, 2019.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[8] G. O. Mohler, M. B. Short, S. Malinowski, M. Johnson, G. E. Tita, A. L. Bertozzi, and P. J. Brantingham, "Randomized controlled field trials of predictive policing," *Journal of the American Statistical Association*, vol. 110, no. 512, pp. 1399–1411, 2015.

[9] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011.

[10] B. Green, T. Horel, and A. V. Papachristos, "Modeling Contagion Through Social Networks to Explain and Predict Gunshot Violence in Chicago, 2006 to 2014," *JAMA Internal Medicine*, vol. 177, no. 3, pp. 326–333, 03 2017.

[11] H.-W. Kang and H.-B. Kang, "Prediction of crime occurrence from multi-modal data using deep learning," *PloS one*, vol. 12, no. 4, p. e0176244, 2017.

[12] B. Wang, P. Yin, A. L. Bertozzi, P. J. Brantingham, S. J. Osher, and J. Xin, "Deep learning for real-time crime forecasting and its ternarization," *Chinese Annals of Mathematics, Series B*, vol. 40, no. 6, pp. 949–966, 2019.

[13] G. Mohler, M. Porter, J. Carter, and G. LaFree, "Learning to rank spatio-temporal event hotspots," *Crime Science*, vol. 9, no. 1, pp. 1–12, 2020.

[14] G. Mohler and M. D. Porter, "Rotational grid, pai-maximizing crime forecasts," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 11, no. 5, pp. 227–236, 2018.

[15] A. Khosla, B. An An, J. J. Lim, and A. Torralba, "Looking beyond the visible scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3710–3717.

[16] A. Bogomolov, B. Lepri, J. Staiano, N. Oliver, F. Pianesi, and A. Pentland, "Once upon a crime: towards crime prediction from demographics and mobile data," in *Proceedings of the 16th international conference on multimodal interaction*, 2014, pp. 427–434.

[17] R. Al-Zaidy, B. C. Fung, A. M. Youssef, and F. Fortin, "Mining criminal networks from unstructured text documents," *Digital Investigation*, vol. 8, no. 3, pp. 147 – 160, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287612000023

[18] S. Seo, H. Chan, P. J. Brantingham, J. Leap, P. Vayanos, M. Tambe, and Y. Liu, "Partially generative neural networks for gang crime classification with partial information," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 257–263.

[19] A. Karazeev. Generative adversarial networks (GANs): Engine and applications.

[20] C. Sornsoontorn. How do GANs intuitively work? [Online]. Available: https://hackernoon.com/how-do-gans-intuitively-work-2dda07f247a1

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015.

[22] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," 2016.

[23] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015.

[24] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," 2016.

[25] Z. Liu, J. Wang, and Z. Liang, "Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation," 2019.

[26] T. Che, Y. Li, R. Zhang, R. D. Hjelm, W. Li, Y. Song, and Y. Bengio, "Maximum-likelihood augmented discrete generative adversarial networks," 2017.

[27] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," 2016.

[28] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin, "Adversarial feature matching for text generation," 2017.

[29] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. null, p. 723–773, Mar. 2012.

[30] A. Lamb, A. Goyal, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," 10 2016.

[31] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," 2017.

[32] K. Lin, D. Li, X. He, Z. Zhang, and M.-t. Sun, "Adversarial ranking for language generation," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3155–3165.

[33] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[34] S. Lu, Y. Zhu, W. Zhang, J. Wang, and Y. Yu, "Neural text generation: Past, present and beyond," 2018.

[35] Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu, "Texygen," *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '18*, 2018.

[36] S. Syed and M. Spruit, "Full-text or abstract? examining topic coherence scores using latent dirichlet allocation," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2017, pp. 165–174.

[37] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.