applied sciences

MDPI

Article

OMECDN: A Password-Generation Model Based on an Ordered Markov Enumerator and Critic Discriminant Network

Jihan Jiang, Anmin Zhou, Liang Liu and Lei Zhang *

School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China
* Correspondence: zhanglei2018@scu.edu.cn

Abstract: At present, static text passwords are still the most widely-used identity authentication method. Password-generation technology can generate large-scale password sets and then detect the defects in password-protection mechanisms, which is of great significance for evaluating password-guessing algorithms. However, the existing password-generation technology cannot ignore low-quality passwords in the generated password set, which will lead to low-efficiency password guessing. In this paper, a password-generation model based on an ordered Markov enumerator and critic discriminant network (OMECDN) is proposed, where passwords are generated via an ordered Markov enumerator (OMEN) and a discriminant network according to the probability of the combination of passwords. OMECDN optimizes the performance of password generation with a discriminative network based on the good statistical properties of OMEN. Moreover, the final password set is formed by the selected passwords with a higher score than the preset threshold, which guarantees the superiority of the hit rate of almost all ranges of combinations of passwords over the initial password set. Finally, the experiments show that OMECDN achieves a qualitative improvement in hit rate metrics. In particular, regarding the generation of 10^7 passwords on the RockYou dataset, the matching entries of the password set generated by the OMECDN model are 25.18% and 243.58% higher than those generated by the OMEN model and the PassGAN model, respectively.

Keywords: password generation; ordered Markov enumerator; discriminant network; password sets; password selection

Citation: Jiang, J.; Zhou, A.; Liu, L.; Zhang, L. OMECDN: A Password-Generation Model Based on an Ordered Markov Enumerator and Critic Discriminant Network. *Appl. Sci.* **2022**, *12*, 12379. https://doi.org/10.3390/app122312379

Academic Editors: Petr Dzurenda, Sara Ricci and Jordi Castellà-Roca

Received: 21 October 2022
Accepted: 29 November 2022
Published: 3 December 2022

1. Introduction

At present, password-based identity authentication is one of the most popular authentication methods; however, the use of passwords is not always secure. Generally, people are used to setting short passwords or character sequences that are easy to remember [1,2], such as "123456". In addition, due to the endowment effect [3], even people with high-security Markov awareness still use weak passwords. In this regard, many researchers have proposed password setting strategies [4–8] to help users set passwords that are easy to remember but difficult to guess.

Research on password strategy and password security, such as detecting the defects of existing user password-protection mechanisms, requires large-scale password plaintext samples. However, it is difficult to completely describe the real-world password plaintext situation accurately. Therefore, password-generation technology to generate large-scale password sets is currently a widely-applied method.

In the available studies [9,10], a situation can be observed where models based on statistical methods can generate a password set according to the usage probability of passwords and place the more commonly-used passwords at the front to speed up password cracking, which does not accurately reflect the distribution feature of the original password set when the generated quantity is large.

Although models based on neural networks can accurately reflect the distribution characteristics of the original password set, the performance of the password generator decreases with the increase of training rounds [11], which means that the repetition rate of passwords becomes higher. These two types of situations will reduce the quality of the password set and further reduce the efficiency of password cracking engineering. In particular, it is observed that the advantages of these two types of models are complementary.

Aiming at the above-mentioned problems, we considered that, if two models with different characteristics can be integrated, the impacts of these shortcomings will be weakened or even eliminated, which means the password-generation model itself will be greatly improved. Accordingly, a password-generation model based on as ordered Markov enumerator and critic discriminant network (OMECDN) is proposed, where passwords are generated via an ordered Markov enumerator (OMEN) and a discriminant network according to the probability of the combination of passwords. First, the OMEN model generates a password set. Passwords are combined, scored, and filtered by a discriminative neural network. Then, the final password set is formed by the selected passwords with a higher score than a preset threshold.

For instance, in the case of the generation of 10^7 passwords, the matching entries of the password set generated by the OMECDN model on the RockYou dataset were 25.18% higher than those of the OMEN model and 243.58% higher than those of the PassGAN model, respectively. The main work and contributions are as follows:

(1) A password generation selection model is proposed, which is based on an ordered Markov enumerator and a critic discriminant network. Accordingly, the password set generated by this model has features of both the above methods: it can be ranked according to the password combination probability, and it can match the password distribution of the real password set.
(2) Compared with the existing GAN-based methods, OMECDN sorts according to the probability of password combination, which can remedy the problem of the repetition rate.
(3) Experimental results show that OMECDN achieved a qualitative improvement in hit rate metrics compared to its base model.

2. Related Work

Password guessing research can be divided into online guessing and offline guessing according to whether it interacts with the server, and wandering guessing and targeted guessing according to whether the user's personal information is used in the guessing process. The current mainstream wandering-guessing algorithms include Markov-based algorithms, PCFG-based algorithms, and neural-network-based algorithms.

In 2005, Narayanan et al. [12] used the Markov model to model a password dataset, which laid the foundation for follow-up research. In 2009, Weir et al. [13] proposed Probabilistic Context-Free Grammar (PCFG). This research provided the foundation for subsequent research on password templates.

Neural-network-based password generation research began earlier; however, it was slow to develop and did not receive much attention [14]. It was not until 2016 that the use of neural networks to research password security became a focus. In 2016, Melicher et al. [15] used recurrent neural network technology to conduct password generation and password strength evaluation research, and subsequent studies [16–18] were based on this research.

In 2019, Hitaj et al. [11] proposed a PassGAN neural network password-guessing method based on a generative adversarial neural network, and subsequent studies [19,20] were based on this model. Zhang et al. [21] used multiple criteria to evaluate the performance of multiple password guessing attack models. Experimental data showed that, when the number of guesses was the same, the password-guessing method combined by the two methods performed better than a single method. Combination method password guessing research has also become a focus of current research.

The above-mentioned password-generation model belongs to the category of wandering attacks, and the attacker does not conduct password guessing attacks against specific users. Another type of password-generation method focuses on specific user information. The attack scenario is often an online password guessing attack, with restrictions, such as the number of guesses. It is more stringent and belongs to the category of online targeted attacks as documented in [22–24] Password-generation model.

Regarding previous password generation research, [9–11], in this paper, we highlight and analyze the ordered Markov enumerator (OMEN) model [9] and the PassGAN model [11], which are password-generation methods based on traditional Markov chains and a generative adversarial network (GAN) [25], respectively. The starting point of the OMEN model is that high-probability passwords are ranked in front of the password set to accelerate password cracking.

The experiments show that the passwords generated by OMEN are arranged in the order of the combined probability of the character sequence. In addition, the passwords generated by the PassGAN model possess the distribution features of the original password set. Based on the existing research, we attempt to find a better password-generation method, e.g., to analyze the complementary advantages that can be better utilized.

3. Password Generation Selection Model

3.1. Overview of Method

We built a model from the perspective of filtering low-quality passwords (belonging to the category of offline walk attacks). This is because existing password-generation models use or combine multiple techniques to generate large-scale password sets, which also produce a large number of low-quality passwords (which makes password guessing less efficient).

A password generation and selection model was instantiated and named the OMECDN model. The schematic diagram of OMECDN model training and password generation is shown in Figure 1.

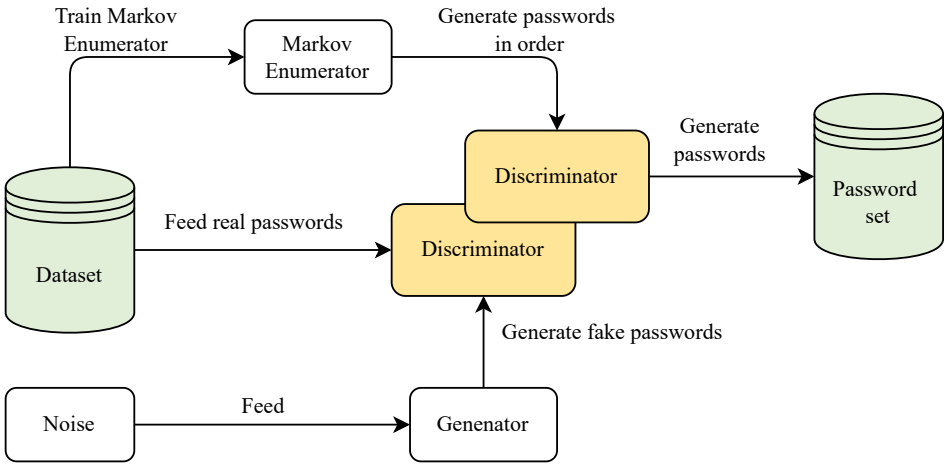

Figure 1. OMECDN framework.

As shown in Figure 1, the PassGAN neural network and the ordered Markov enumerator are trained with the same dataset. The OMECDN model uses the discriminator of the generative adversarial network as the password selection module to implement the algorithm, and the generator network is discarded. We use the ordered Markov enumerator to generate candidate password sets and apply the discriminator network to the candidate password set generated by the ordered Markov password enumerator to obtain the corresponding scores. Using the OMECDN model selection score, the passwords whose scores are greater than or equal to the threshold constant constitute the final password set.

3.2. Password Generation Selection Model

The ordered Markov enumerator (OMEN) was chosen as the password generation module to implement the algorithm. To better understand the generation process of the candidate password set, the relevant theoretical knowledge is supplied. OMEN generates a combined password according to the probability of occurrence of multiple N-gram character sequences. The higher the probability, the higher the output of the combined password. The specific method is to use Equation (1) to convert all N-gram sequence probabilities p_i into discrete integers $level_i$, a fixed interval

$$level_i = round(\log(c_1 \cdot p_i + c_2)) \tag{1}$$

where c_1 and c_2 are selected constants and round is the rounding function.

The concept of storing N-gram sequence heaps with the same level value is introduced, and the N-gram sequences are stacked according to the *level* value [9]. Using select N-gram sequences from different heaps to form password X makes the sum of the level values of the combined password equal to $H(X)$ and the length of the combined password equal to l. Adjusting $H(X)$ and l can achieve the purpose of controlling the output of the generated combined password on the order of the probability of the combined password. The OMEN model generated password set D_{OMEN} can be expressed as

$$D_{\text{OMEN}} = \left\{ X : \|X\| \in \mathbb{Z}^{[0,L]}, H(X) \in \prod [level_0, level_{\text{all}}] \right\} \tag{2}$$

where $\|X\| \in \mathbb{Z}^{[0,L]}$ is the combined password length and $H(X)$ is the sum of the level value of the combined password. Assuming that the maximum value of the generated password set length l is L, the sum of all *level* values is $level_{\text{all}}$, and the minimum *level* is $level_0$.

The process of the PassGAN neural network discriminator for the candidate password set password X generated by OMEN to calculate the score S can be described as

$$S = f_{\text{GAN-D}}(X) \tag{3}$$

where $f_{\text{GAN-D}}$ is the PassGAN discriminator neural network.

Combining Equations (2) and (3), the password set D_{OMECDN} generated by the OMECDN model is expressed as

$$D_{\text{OMECDN}} = \{ X : X \in D_{\text{OMEN}}, f_{\text{GAN-D}}(X) \geq \gamma \} \tag{4}$$

where the range of X is referenced to Equation (2), the constant γ is determined experimentally (more details about γ are described in Algorithm 1), and the number of generated passwords for the target is prepared.

Algorithm 1: The OMECDN model generation password algorithm

```
Count = 0
    WHILE Count < targetCount:
        FOR each Password X IN D_OMEN:
            IF f_GAN-D(X) ≥ γ:
                Output X
                Count = Count+1
            ELSE:
                CONTINUE
            END IF
        END FOR
    END WHILE
```

The trained ordered Markov password enumerator is used to generate candidate password sets. The generation process is to traverse each heap according to the level value and find substring sequences from these heaps to combine passwords to generate combined passwords. The calculation of the password score is shown in Equation (3), and the process of selecting a password based on the score is shown in Figure 2.

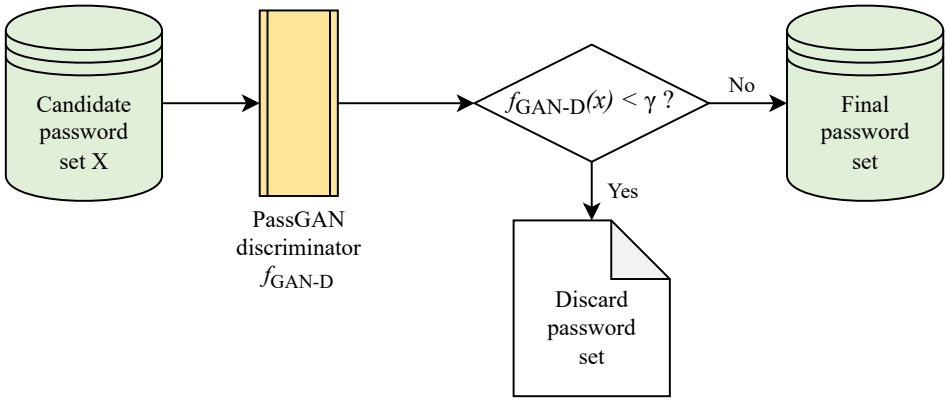

Figure 2. The password choosing procedure of OMECDN.

The candidate password set X is used as the input of the PassGAN model discriminator $f_{\text{GAN}-\text{D}}$ to obtain the score S. The score S and the threshold constant γ are judged, and the passwords greater than or equal to the threshold constant γ are selected to form the final password set.

4. Experiment Design

This section contains three parts to evaluate the principles, data, and preparatory experiments. Password generation experiments are performed on the OMEN model and PassGAN model, and some of the features they exhibit are analyzed to better understand the comparison objects of the experiments in the evaluation experiments in the next section.

4.1. Principles of Evaluation

The password generation selection instantiation model OMECDN proposed in this chapter belongs to the offline walking password attack model and does not pay attention to specific users and specific identity information.

Evaluating the effectiveness of the OMECDN model is used to calculate the hit rate (*Hit_rate*) of the final password set on the test set. The calculation method is shown as

$$Hit\_rate = \frac{Hit}{Testing\_count} * 100\% \tag{5}$$

where *Hit* is the hit frequency of the generated password set in the test set and *Testing_count* is the number of passwords in the test set. For example, if 100 passwords are generated, 30 will appear in the test set, and there are 1000 passwords in the test set. In this instance, the generated password hit rate is 3%.

The experiment also counts the repetition rate (*Repetition*) of the PassGAN model. The calculation method of the repetition rate is shown as

$$Repetition = \frac{Hit - Hit_{\text{unique}}}{MaxTry} * 100\% \tag{6}$$

where *Hit* is the number of hits of the generated password set in the test set, Hit_{unique} is the number of remaining passwords in the hit test set to delete duplicate passwords, and *MaxTry* is the maximum number of generated passwords. For example, if 100 passwords

are generated, 50 will appear in the test set, and 20 passwords will remain after the repetition is deleted. The password repetition rate generated in this case is 30%.

The OMECDN model method has a password selection process, and the password hit rate (*Hit_rate*) calculation method is calculated as

$$Hit\_rate = \frac{Hit_{f_{GAN-D}(X)} \geq \gamma}{Testing\_count} * 100 \quad (7)$$

In contrast to using Equation (5) to calculate the hit rate, OMECDN only considers passwords whose scores in the candidate password set are greater than or equal to the threshold γ. This decision was determined from the experiments, and the choice of threshold can continue to preserve the well-performing generation results in OMEN. This will be explained in detail in Section 5.

4.2. Data Preprocessing

The password dataset is preprocessed as follows:

(1) Delete repeated passwords in the dataset.
(2) Delete passwords containing characters in the illegal character set in the dataset. The legal character set is "qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890˜!@#$%^&*()_+-=[]{};'":,.<>/?|\".
(3) The PassGAN neural-network model requires 10 characters for the length of the input password sequence; thus, delete passwords with a length greater than 10 in the dataset.
(4) Disrupt the sequence of passwords in the dataset.
(5) After processing the dataset according to the above process, divide each dataset into a training set and a test set according to the ratio of 4:1.

4.3. Environment and Dataset

The experimental software and hardware environment are shown in Table 1.

Table 1. Hardware and software environment.

Operating system	Microsoft Windows 10.0.18363.1082
CPU	Intel Core i7-9700 CPU @ 3.00 GHz 3.00 GHz
GPU	NVIDIA GeForce RTX 2070 SUPER
CUDA version	Release 10.1, V10.1.243
Storage	NVMe GALAX TA1H0240N 240G

A total of four leaked password datasets were collected. Table 2 summarizes the number of passwords contained in these leaked password datasets and the source of the password sets.

Table 2. Leaked password dataset information.

Dataset Name	Passwords	De-Duplicated Passwords	Source
Yahoo	453,492	342,515	Raidforums
12306	131,653	117,768	Raidforums
CSDN	6,428,630	4,034,934	Raidforums
RockYou	32,602,881	14,344,391	Skullsecurity [1]

[1] https://wiki.skullsecurity.org/Passwords, accessed on 15 June 2022.

4.4. Features of OMEN Model

According to the content described in Section 3, we fed the Rockyou training set to the OMEN model for learning. After the N-gram sequence level value was counted, we used

the content to generate a candidate password set, where the N-gram sequence values of N are 2, 3, 4, and 5, respectively. The purpose of this experiment is to search for the best N value and calculate the hit rate of the OMEN model on the Rockyou test set. The hit rate is shown in Figure 3.

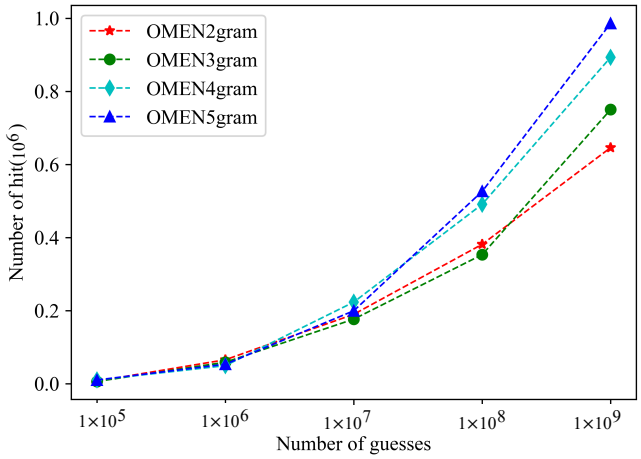

Figure 3. The hits of OMEN models.

In the legend in Figure 3, OMEN2gram indicates that N in the N-gram of the OMEN model is set to 2. Other legends are similar. The experimental results show that, when N in the OMEN model N-gram was 5, the OMEN model had the best hit performance.

Blase Ur et al. [26] argued that password-guessing algorithms based on Markov models are more efficient than PCFG-based guessing algorithms for cracking. For important improvements of the PCFG approach, such as the GENPass [27] model (PCFG + LSTM (PL) approach), there are no relevant experiments to reproduce the specific process, and thus no comparison with the work in this paper is reflected.

4.5. Features of PassGAN

The PassGAN neural-network model was trained with the same Rockyou training set of the trained ordered Markov enumerator. The experiment changed the maximum character sequence length (Sequence_length) and the number of training rounds (Iterations) for comparison experiments. The hit situation of the PassGAN neural-network model is shown in Figure 4.

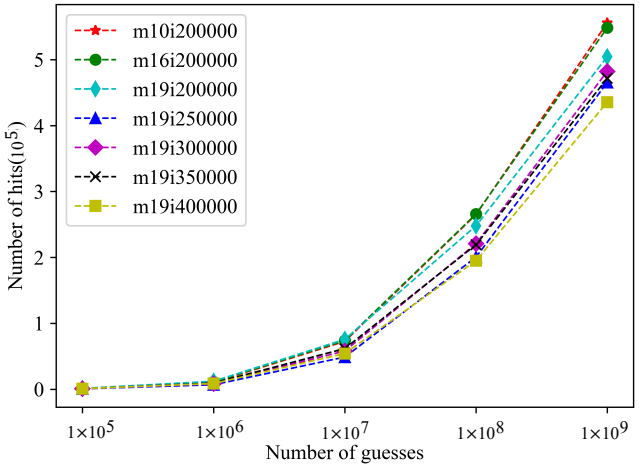

Figure 4. The hits of PassGAN models.

In the legend in Figure 4, m10i200000 means that the maximum sequence length (Sequence_length) of the PassGAN neural-network model is 10, and the maximum number

of training rounds (iterations) is 200,000. Other legends are similar. The results show that, when the maximum sequence length of the PassGAN neural-network model is 10, and the maximum number of training rounds is 200,000, the PassGAN neural-network model has the best hit performance.

Combining Figures 3 and 4, it can be seen that the PassGAN model hit rate is worse than the OMEN model. One of the reasons is that the PassGAN model generates duplicate passphrases. Therefore, according to Equation (5), the PassGAN model duplication rate is statistically analyzed, and the results are shown in Figure 5.

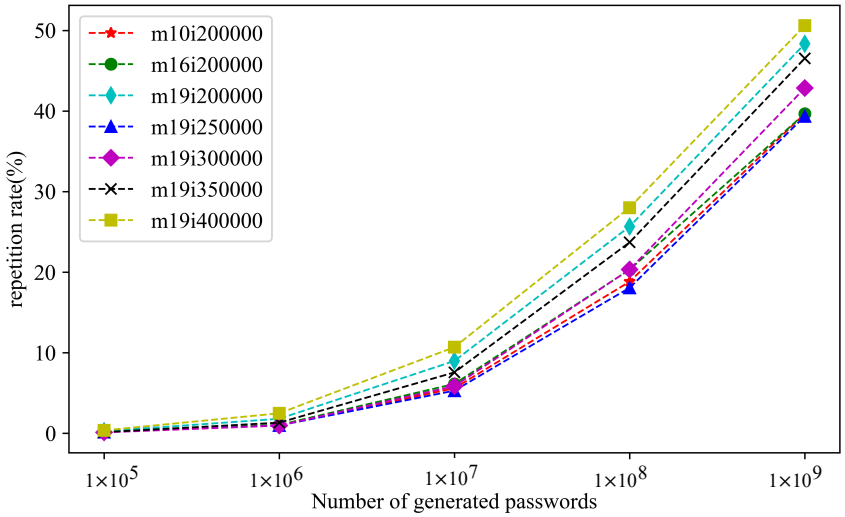

Figure 5. Effects of training rounds on the repetition rate of PassGAN.

Combining Figures 4 and 5, the following conclusions are drawn. When the number of training rounds was taken as 400,000 and the maximum character sequence length was 19, the password repetition rate reached 50.60% in the case of 10^9 generated passwords. The PassGAN model reflects the feature that the password repetition rate increases with the number of generated passwords for seven different parameter cases. However, the performance of the PassGAN model generator does not increase but becomes worse as the number of training rounds and the maximum character sequence length increase, while the repetition rate increases with the number of training rounds.

In fact, the best performance of the generator was reached at about 200,000 training rounds, and this decreased as the number of training rounds rose. In the subsequent training, the discriminator has the advantage of obtaining a network with a similar structure to the generator (which better reproduces the distribution of the original dataset) while discarding the function of generating duplicate passphrases.

5. Results

Through the preparatory experiments in the previous section, some features in the OMEN model and PassGAN model are verified, which aids in understanding the experiments on OMECDN in this section.

Similarly, hit rate test experiments were conducted on OMECDN to observe how OMECDN performs under different datasets and concerning different models, and the conclusions indicate how much superiority OMECDN has compared to the reference experiments. Further, the implications of the threshold selection in OMECDN are analyzed to verify the feasibility of the OMECDN screening process.

5.1. OMECDN Model Evaluation

We used the OMEN-5gram model trained in Section 4 as the password generation module implementation algorithm in the password generation selection model and the PassGAN neural-network model discriminator trained in Section 4 as the password selec-

tion module implementation algorithm in the password generation selection model to form the OMECDN model. The score thresholds γ were set to -1.2, -1.3, and -1.4, respectively. According to Algorithm 1, the password set was generated, and the password set generated by the OMECDN model was compared with the Rockyou test set according to Equation (7). The results are shown in Figure 6.

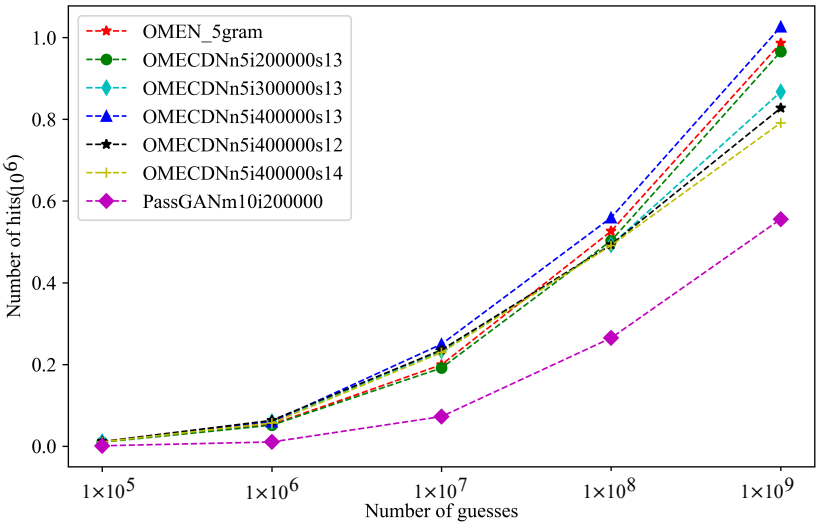

Figure 6. The hit rate amounts for the three models.

In the legend in Figure 6, OMECDNn5i200000s13 represents the implementation of the password generation module of the OMECDN model. In the N-gram of OMEN, N is 5, the number of training rounds is 200,000, and the score threshold constant γ is -1.3. The other legends are also the same. The results show that, when the number of training rounds was 400,000 and the score threshold constant γ was set to -1.3, the OMECDN model had the highest hit rate when the same number of password sets were generated. The specific numbers of hits of the three models are shown in Table 3.

Table 3. Hits of the three models.

Guesses	Models			Improvement	
	OMEN (*Hit_Rate*)	**PassGAN (*Hit_Rate*)**	**OMECDN (*Hit_Rate*)**	**(OMECDN-OMEN)/OMEN $*$ 100%**	**(OMECDN-PassGAN)/ PassGAN $*$ 100%**
10^5	10,045 (0.51%)	1146 (0.06%)	11,272 (0.57%)	12.22%	883.60%
10^6	53,257 (2.69%)	10,539 (0.53%)	60,123 (3.04%)	12.89%	470.48%
10^7	199,507 (10.09%)	72,691 (3.68%)	249,750 (12.63%)	25.18%	243.58%
10^8	526,296 (26.62%)	264,978 (13.40%)	559,439 (28.29%)	6.30%	111.13%
10^9	986,322 (49.88%)	555,724 (28.10%)	1,026,474 (51.91%)	4.07%	84.71%

Using the same experimental method, the password generation fitting experiment was performed on the Yahoo, 12306, and CSDN datasets. In the case of generating 10^5 passwords, the password results are shown in Figure 7.

The experimental data of password generation and fitting of different datasets show that, in the Rockyou, 12306, and CSDN datasets, OMECDNn5i200000s13 had the highest hit rate, and, in the Yahoo dataset, OMECDNn5i200000s12 had the highest hit rate. In summary, the OMECDN model performed better than the OMEN model and the PassGAN model. The network model is a better password-generation model.

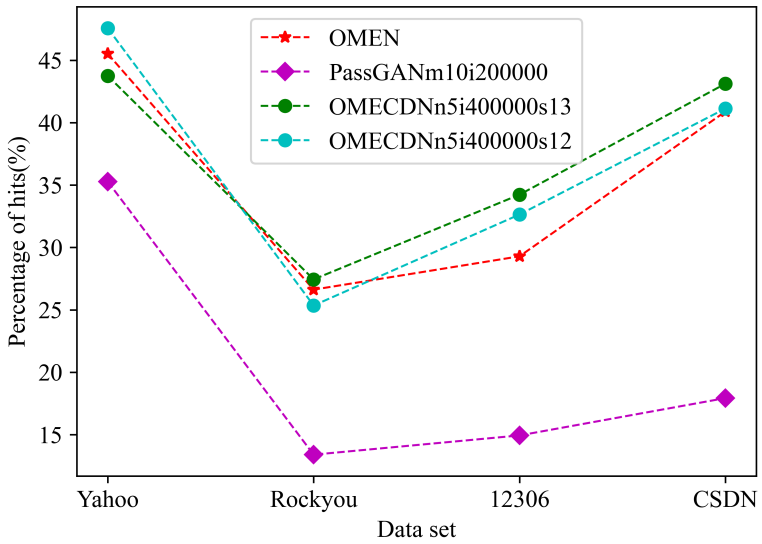

Figure 7. The hit rate amounts of the three hit models for four datasets.

5.2. OMECDN Model Score Distribution

In Section 3.2, we introduced that OMEN realizes the generation of combined passwords based on the probability of multiple N-gram character sequences (the method is to convert all N-gram sequence probabilities p into discrete fixed intervals). Understanding this section of the interval helps to read the subsequent figures.

In this section, we analyze the distribution of teh password scores in the password set generated by the password generation module algorithm in the OMECDN model. The password scores are divided into 41 intervals, which are:

$$(-\infty, -1.9], (-1.9, -1.8], (-1.8, -1.7], \ldots, (1.8, 1.9], (1.9, 2.0], (2.0, +\infty)$$

The first 40 intervals are half-open intervals. Except for the first interval and the last interval, the length of each interval is 0.1. The OMECDN model counts and classifies the passwords generated by OMEN into 41 intervals, and then calculates the efficiency of the OMECDN model ($Efficiency_{\text{OMECDN}}$) for each interval. The efficient calculation method is shown in Equation (8).

$$Efficiency_{\text{OMECDN}} = \frac{Hit_{f_{\text{GAN-D}}(X)} \geq -1.3}{MaxTry_{f_{\text{GAN-D}}(X)} \geq -1.3} * 100\% \tag{8}$$

Furthermore, the calculation method of the efficiency of the OMEN model ($Efficiency_{\text{OMEN}}$) is shown in Equation (9).

$$Efficiency_{\text{OMEN}} = \frac{Hit}{MaxTry} * 100\% \tag{9}$$

In Equation (8), $Hit_{f_{\text{GAN-D}}(X)} \geq -1.3$ represents the number of passwords in the test set generated by the OMECDN model, and $MaxTry_{f_{\text{GAN-D}}(X)} \geq -1.3$ represents the number of passwords in the final password set of the OMECDN model. In Equation (9), Hit is the number of hit passwords in the test set for the generated password set for the OMEN model, and $MaxTry$ is the total number of passwords generated for the OMEN model. The statistical results of the password score distribution are shown in Figures 8–11.

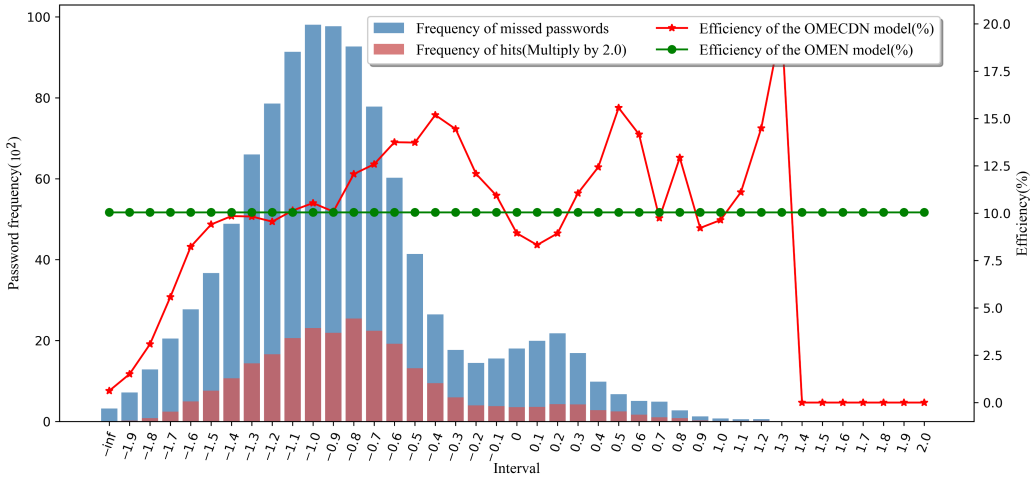

Figure 8. The distribution of 1×10^5 passwords scores generated by the OMECDN model.

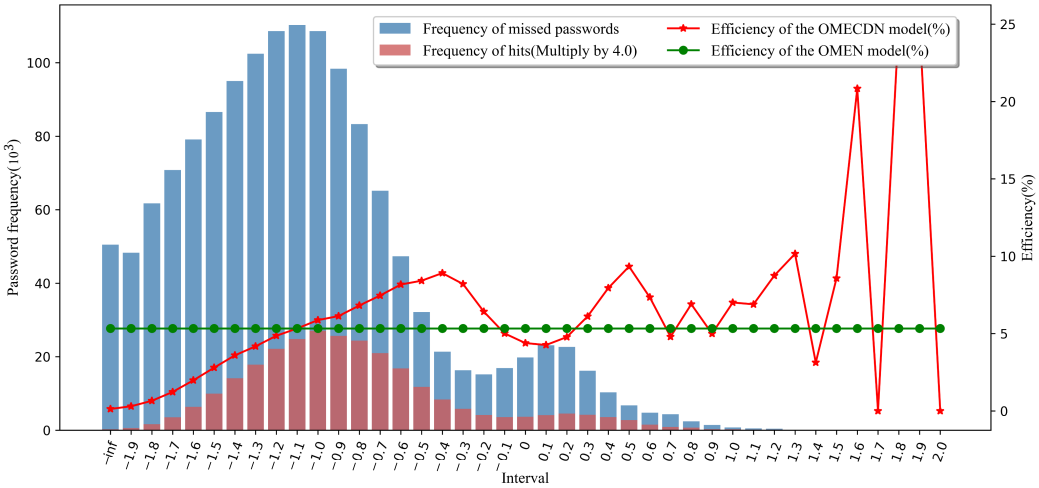

Figure 9. The distribution of 1×10^6 passwords scores generated by the OMECDN model.

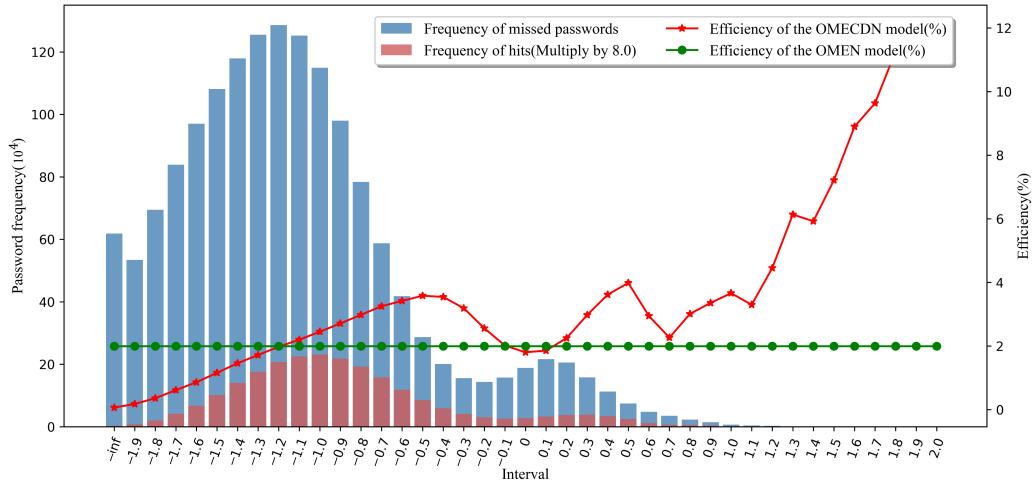

Figure 10. The distribution of 1×10^7 passwords scores generated by the OMECDN model.

In the password score distributions shown in Figures 8–11, the green line indicates the efficiency of the OMEN model in generating the same number of passwords under

the same training set and test set. The red line represents the efficiency of the OMECDN model in different intervals when the same number of passwords are generated under the same training set and test set. To better visualize the effects in the figures, we enlarged the columns for the data of the frequency of hitting passwords.

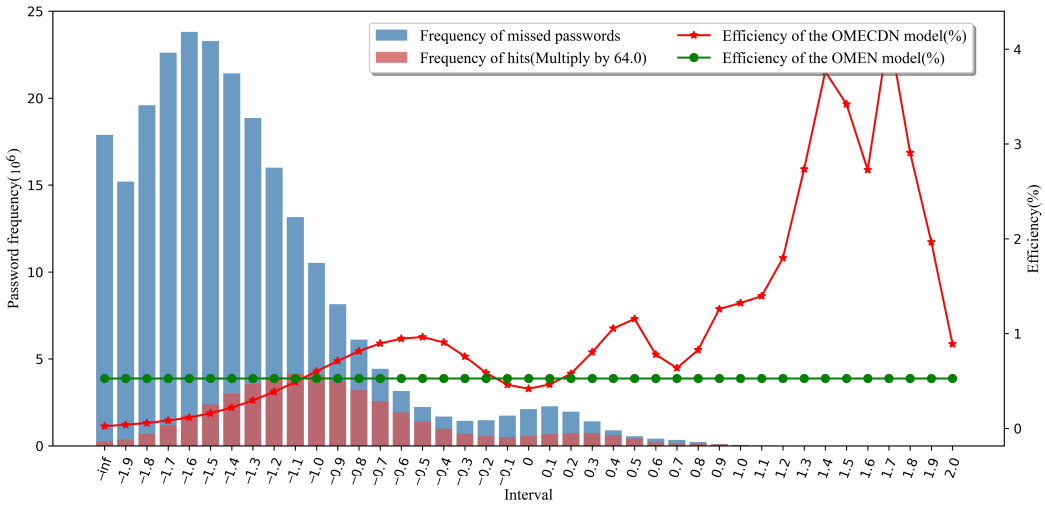

Figure 11. The distribution of 1×10^8 passwords scores generated by the OMECDN model.

To facilitate readers to better understand these figures, we take Figure 8 as an example for analysis. The abscissa represents the interval (it can be simply understood as these discrete intervals corresponding to or correlating with the probability of the N-gram character sequence in the Markov model). The side ordinate represents the password frequency, the red bar corresponds to the password frequency of the OMECDN model in the current interval, the blue bar corresponds to the number of missed passwords in the current interval, and the sum of the heights of the red and blue bars is the current experiment.

The total number of password sets is 10^5 (for better visualization, we enlarged the height of the red column accordingly, which does not affect the conclusion, and this is enlarged by two times in Figure 8). After understanding the bar graph to indicate the distribution of password scores, we continue to analyze the line graph. The ordinate on the right represents the efficiency of the password guessing model. The calculation method was introduced above. The green line corresponds to the OMEN model (as a reference). When the red line is above the green line, it means that the OMECDN model is more efficient than the OMEN model. It can be seen in Figure 9 that, on the right side of the interval "−1.1", the efficiency of the OMECDN model is almost better than that of the OMEN model.

From the password score distribution diagram, the PassGAN neural network discriminator has a certain rule for the password output score distribution, and the hit password score distribution is concentrated around −1.0 and 0.3. When it is greater than or equal to the threshold constant −1.3, the OMECDN model is more efficient. Some intervals are higher than the overall efficiency of the OMEN model, and thus the overall efficiency of the OMECDN model is higher than the overall efficiency of the OMEN model. This proves that an appropriate threshold constant was selected, and the performance of the OMECDN model is better than that of the OMEN model. In a real-world scenario, the threshold can be chosen flexibly to ensure the highest possible cracking efficiency. This also proves the validity and feasibility of the password generation selection model.

6. Conclusions

The use of password-generation technology to generate large-scale password sets, to detect the defects of existing user password-protection mechanisms, and to evaluate the efficiency of password-guessing algorithms is important for studying password security. Aiming at the research of password-generation technology, a password-generation model

based on an ordered Markov enumerator and critic discriminant network (OMECDN) was proposed in this paper. OMECDN optimizes the performance of password generation with a discriminative network based on the good statistical properties of OMEN. Moreover, the final password set was formed by the selected passwords with a higher score than the preset threshold, which guarantees the superiority of the hit rate of almost all ranges of combinations of passwords over the initial password set.

When 10^9 passwords were generated, the hit rate reached 51.91%, which is 4.07% higher than the OMEN model and 84.71% higher than the PassGAN model. In the case of generating 10^7 passwords, the password hit entries generated by the OMECDN model were 25.18% higher than the password hit entries generated by the OMEN model and 243.58% higher than the password hit entries generated by the PassGAN model. The OMECDN model is better than the OMEN model and the PassGAN model and is a better password-generation model.

Future work can focus on the distribution of the score output by the neural-network module. Studying the output of the neural-network module can discover the internal working mode of the neural network and determine what kind of structure of password score value can exceed the threshold. This will have a positive effect on improving the research of password-construction strategies.

Author Contributions: Conceptualization, L.Z.; methodology, J.J.; software, J.J.; validation, J.J. and L.L.; formal analysis, J.J.; investigation, L.L.; resources, L.Z.; data curation, L.L.; writing—original draft preparation, J.J.; writing—review and editing, J.J.; visualization, L.L.; supervision, A.Z.; project administration, A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Sichuan Science and Technology Program under Grant 2021YFG0159 & 2022YFG0171, in part by the Fundamental Research Funds for the Central Universities 2021SCU12136.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, D.; Wang, P.; He, D.; Tian, Y. Birthday, name and bifacial-security: Understanding passwords of Chinese web users. In Proceedings of the 28th USENIX Security Symposium (USENIX Security 19), Santa Clara, CA, USA, 14–16 August 2019; pp. 1537–1555.
2. Zeng, J.; Duan, J.; Wu, C. Empirical study on lexical sentiment in passwords from Chinese websites. *Comput. Secur.* **2019**, *80*, 200–210. [CrossRef]
3. Renaud, K.; Otondo, R.; Warkentin, M. "This is the way 'I'create my passwords"…does the endowment effect deter people from changing the way they create their passwords? *Comput. Secur.* **2019**, *82*, 241–260. [CrossRef]
4. Guo, Y.; Zhang, Z.; Guo, Y. Optiwords: A new password policy for creating memorable and strong passwords. *Comput. Secur.* **2019**, *85*, 423–435. [CrossRef]
5. Siponen, M.; Puhakainen, P.; Vance, A. Can individuals' neutralization techniques be overcome? A field experiment on password policy. *Comput. Secur.* **2020**, *88*, 101617. [CrossRef]
6. Ye, B.; Guo, Y.; Zhang, L.; Guo, X. An empirical study of mnemonic password creation tips. *Comput. Secur.* **2019**, *85*, 41–50. [CrossRef]
7. Yıldırım, M.; Mackie, I. Encouraging users to improve password security and memorability. *Int. J. Inf. Secur.* **2019**, *18*, 741–759. [CrossRef]
8. Doucek, P.; Pavlíček, L.; Sedláček, J.; Nedomova, L. Adaptation of password strength estimators to a non-English environment—The Czech experience. *Comput. Secur.* **2020**, *95*, 101757. [CrossRef]
9. Dürmuth, M.; Angelstorf, F.; Castelluccia, C.; Perito, D.; Chaabane, A. OMEN: Faster password guessing using an ordered markov enumerator. In Proceedings of the International Symposium on Engineering Secure Software and Systems, Milan, Italy, 4–6 March 2015; pp. 119–132.
10. Linghu, Y.; Li, X.; Zhang, Z. Deep Learning vs. Traditional Probabilistic Models: Case Study on Short Inputs for Password Guessing. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, New York, NY, USA, 2–4 October 2019; pp. 468–483.

11. Hitaj, B.; Gasti, P.; Ateniese, G.; Perez-Cruz, F. Passgan: A deep learning approach for password guessing. In Proceedings of the International Conference on Applied Cryptography and Network Security, Bogotá, Colombia, 5–7 June 2019; pp. 217–237.
12. Narayanan, A.; Shmatikov, V. Fast dictionary attacks on passwords using time-space tradeoff. In Proceedings of the 12th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 7–11 November 2005; pp. 364–372.
13. Weir, M.; Aggarwal, S.; De Medeiros, B.; Glodek, B. Password cracking using probabilistic context-free grammars. In Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 17–20 May 2009; pp. 391–405.
14. Ciaramella, A.; D'Arco, P.; De Santis, A.; Galdi, C.; Tagliaferri, R. Neural network techniques for proactive password checking. *IEEE Trans. Dependable Secur. Comput.* **2006**, *3*, 327–339. [CrossRef]
15. Melicher, W.; Ur, B.; Segreti, S.M.; Komanduri, S.; Bauer, L.; Christin, N.; Cranor, L.F. Fast, lean, and accurate: Modeling password guessability using neural networks. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 175–191.
16. Xu, R.; Chen, X.; Shi, J. A coarse-grained password model with memorable unit-based recurrent neural networks. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 1890–1897.
17. Zhang, M.; Zhang, Q.; Hu, X.; Liu, W. A Password Cracking Method Based On Structure Partition and BiLSTM Recurrent Neural Network. In Proceedings of the Eighth International Conference on Communication and Network Security, Qingdao, China, 2–4 November 2018; pp. 79–83.
18. Fang, Y.; Liu, K.; Jing, F.; Zuo, Z. Password guessing based on semantic analysis and neural networks. In Proceedings of the Chinese Conference on Trusted Computing and Information Security, Wuhan, China, 18 October 2018; pp. 84–98.
19. Nam, S.; Jeon, S.; Moon, J. A new password cracking model with generative adversarial networks. In Proceedings of the International Workshop on Information Security Applications, Jeju Island, Republic of Korea, 21–24 August 2019; pp. 247–258.
20. Nam, S.; Jeon, S.; Kim, H.; Moon, J. Recurrent gans password cracker for iot password security enhancement. *Sensors* **2020**, *20*, 3106. [CrossRef] [PubMed]
21. Zhang, J.; Yang, C.; Zheng, Y.; You, W.; Su, R.; Ma, J. A preliminary analysis of password-guessing algorithm. In Proceedings of the 2020 29th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 3–6 August 2020; pp. 1–9.
22. Wang, D.; Zhang, Z.; Wang, P.; Yan, J.; Huang, X. Targeted online password guessing: An underestimated threat. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1242–1254.
23. Xie, Z.; Zhang, M.; Yin, A.; Li, Z. A new targeted password guessing model. In Proceedings of the Australasian Conference on Information Security and Privacy, Wollongong, Australia, 28–30 November 2020; pp. 350–368.
24. Li, Z.; Li, T.; Zhu, F. An Online Password Guessing Method Based on Big Data. In Proceedings of the 2019 Third International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, Male, Maldives, 23–24 March 2019; pp. 59–62.
25. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2661.
26. Ur, B.; Segreti, S.M.; Bauer, L.; Christin, N.; Cranor, L.F.; Komanduri, S.; Kurilova, D.; Mazurek, M.L.; Melicher, W.; Shay, R. Measuring {Real-World} Accuracies and Biases in Modeling Password Guessability. In Proceedings of the 24th USENIX Security Symposium (USENIX Security 15), Washington, DC, USA, 12–14 August 2015; pp. 463–481.
27. Liu, Y.; Xia, Z.; Yi, P.; Yao, Y.; Xie, T.; Wang, W.; Zhu, T. GENPass: A general deep learning model for password guessing with PCFG rules and adversarial generation. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.