PassGan Red generativa adversaria para generar contraseñas

Ulises Díez Santaolalla
Mathematical Engineering and AI
Pontifical University Comillas
ulises.diez@alu.comillas.edu

Teresa Franco Corzo
Mathematical Engineering and AI
Pontifical University Comillas
tere.franco@alu.comillas.edu

Ignacio Felices Vera
Mathematical Engineering and AI
Pontifical University Comillas
ignacio.felices@alu.comillas.edu

Paloma Díez Amatriaín
Mathematical Engineering and AI
Pontifical University Comillas
paloma.diez@alu.comillas.edu

Abstract

En este trabajo estudiamos PassGAN, una red generativa adversaria entrenada sobre conjuntos de contraseñas filtradas públicamente, con el objetivo de analizar su capacidad para modelar la distribución real de contraseñas humanas y las implicaciones de seguridad que ello conlleva. Describimos la arquitectura y el proceso de entrenamiento, y evaluamos la calidad y diversidad de las contraseñas generadas comparándolas con enfoques clásicos basados en diccionarios y reglas. Nuestros resultados muestran que modelos generativos profundos pueden aproximarse al comportamiento humano y recuperar una fracción significativa de contraseñas con menos intentos, lo que refuerza la necesidad de políticas de autenticación robustas, gestores de contraseñas y mecanismos adicionales como la autenticación multifactor. Finalmente, discutimos las limitaciones del modelo y líneas futuras de trabajo orientadas a utilizar este tipo de técnicas para auditar y mejorar la seguridad de sistemas reales, y no para llevar a cabo ataques.

Contents

1 Introducción

PassGAN es un modelo basado en *Generative Adversarial Networks* (GANs) que aprende patrones contenidos en conjuntos de contraseñas filtradas públicamente con el fin de estudiar la capacidad de los modelos generativos para producir contraseñas plausibles. Este informe consiste en un estudio en profundidad de su invención y usos, junto con una implementación reproducible de PassGAN. Esto permite desarrollar un análisis crítico sobre su eficacia, limitaciones y riesgos en el contexto de la ciberseguridad defensiva.

1.1 Historia

Todo empieza con el artículo *"PassGAN: A Deep Learning Approach for Password Guessing"*, donde los autores combinan las GANs con los *datasets* de contraseñas filtradas públicamente para adivinar contraseñas. En el artículo se demuestra que la arquitectura de las GANs da mejores resultados que los algoritmos del momento, capturando los patrones de las contraseñas.

Las GANs son introducidas por I. Goodfellow en 2014, y consisten en dos partes generales: *discriminador* y *generador* (ver Figura 1). Trabajan juntos, pero con objetivos contrapuestos, de ahí que se denominen "adversarial" (antagónicas). Por un lado, el generador trata de crear nuevos datos muy similares a los reales, mientras que el discriminador trata de identificar el dato generado por el generador. Ambos se entrenan simultáneamente: el generador intenta mejorar su habilidad para engañar al discriminador, y el discriminador intenta mejorar su habilidad para no ser engañado. Esto dura hasta que el generador se vuelve tan bueno que el discriminador ya no puede distinguir entre los datos reales y los generados.

Aunque las GANs se utilizan en una gran variedad de aplicaciones, desde la generación de imágenes hasta la creación de obras de arte, también plantean ciertos desafíos de seguridad, ya que pueden ser utilizadas para crear información falsa o engañosa, como es el caso de las *deepfakes*.

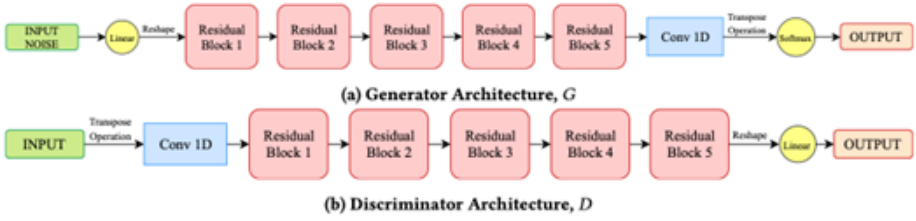


Figure 1: Arquitectura del modelo.

2 Aspectos técnicos

Este tipo de arquitectura se basa en un proceso competitivo entre dos modelos neuronales: un generador (G) y un discriminador (D). Ambos se entrenan simultáneamente en un juego *minimax*, en el que cada red optimiza una función de pérdida opuesta a la de la otra. Para el caso concreto del estudio de contraseñas, esta dinámica adversarial resulta especialmente útil, pues permite que el generador aprenda la distribución estadística de contraseñas reales filtradas y produzca nuevas contraseñas que sean prácticamente indistinguibles para el discriminador.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Figure 2: Función de pérdida del modelo

- **Discriminador** $\rightarrow$ discernir si una contraseña es generada o si pertenece al conjunto de datos reales (maximiza su función de pérdida).
- **Generador** $\rightarrow$ generar contraseñas lo más similares posibles al conjunto de datos reales, cuyo objetivo es minimizar su función de pérdida, es decir, lograr que el discriminador se equivoque lo máximo posible.

El equilibrio óptimo de este juego se alcanza cuando el generador ha aprendido a imitar perfectamente la distribución de los datos reales, de modo que el discriminador no puede diferenciar entre ambas y asigna una probabilidad de 0.5 a todas las muestras.

El objetivo de PassGAN no es memorizar contraseñas exactas del conjunto de entrenamiento, sino aprender la estructura estadística subyacente a millones de contraseñas filtradas previamente. Esto es significativo porque, en la práctica, los usuarios tienden a repetir patrones similares: sustituciones comunes, palabras modificadas, combinaciones de teclado, variantes de una misma raíz léxica, etc.

Por qué esta arquitectura es útil para PassGAN:

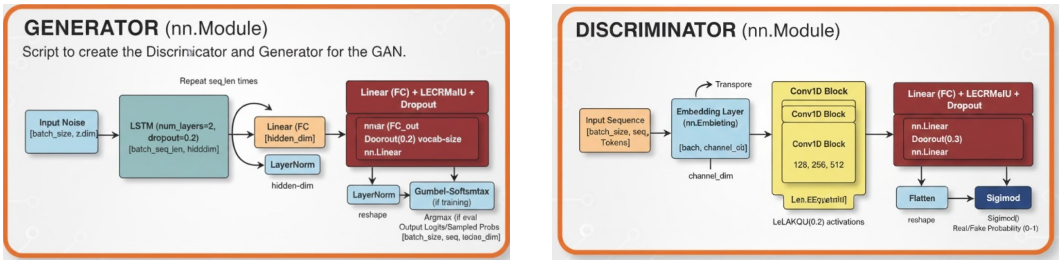
- Aprende sin necesidad de modelar una probabilidad explícita. No trata de aprender una probabilidad, sino de imitarla, lo que ayuda mucho al estar tratando con distribuciones tan complejas (multimodales, etc.).
- Genera muestras verosímiles, no copias del *dataset*.
- Detecta y replica patrones de composición de forma automática.

El objetivo es inferir la distribución estadística que siguen las contraseñas humanas reales. Sirven como una herramienta de estudio tanto para evaluar amenazas como para diseñar mecanismos de defensa más sólidos.

3 Implementación

Se ha considerado relevante realizar un experimento e implementar este algoritmo ya que poseemos los conocimientos para realizarlo. En una aproximación rápida se ha creado un repositorio de GitHub donde se ha entrenado a un modelo GAN a producir contraseñas dado un *dataset* de contraseñas filtradas (nota 1).

Para tratar los datos se ha realizado un diccionario que mapea los caracteres de las contraseñas a índices. Durante el entrenamiento se minimizan dos funciones de pérdida, la del generador y la del discriminador. La del discriminador mide la capacidad de distinguir contraseñas reales de las generadas, y la del generador refleja la habilidad de producir secuencias que engañen al discriminador.



(a) Estructura del generador (b) Estructura del discriminador

Figure 3: Comparación de las estructuras.

Pequeñas oscilaciones en las pérdidas son normales y evidencian el equilibrio dinámico entre generador y discriminador, sin divergencias ni saturaciones, gracias a los ajustes de hiperparámetros.

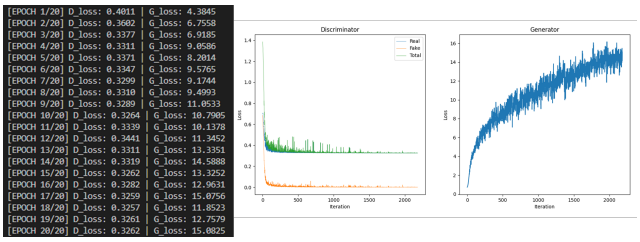


Figure 4: Entrenamiento de la simulación del modelo.

Como podemos ver en las gráficas, los valores de `D_loss` y `G_loss` muestran que el GAN ha aprendido de manera controlada a generar contraseñas plausibles, no simplemente aleatorias. El generador captura patrones y estructuras presentes en el conjunto de datos real, produciendo contraseñas coherentes y diversas, mientras que el discriminador mantiene un equilibrio al discernir ejemplos realistas de irreales.

Esta efectividad se refleja en los `generated_samples.txt` obtenidos tras 20 épocas, usando un tamaño de secuencia de 12, *batch* de 64, un vector de ruido de dimensión 128, capas LSTM de tamaño 256 y 2 niveles, junto con Gumbel-Softmax a temperatura 1.0.

Por ejemplo, algunas contraseñas generadas en la última época son:

- `a$TyaTz)+z#`
- `Jv+@l@@l35_M`
- `RbLFEO^Hb%dK`
- `B3cGE%DlsDA8`

Estos ejemplos muestran que el generador produce contraseñas variadas, combinando letras mayúsculas, minúsculas, números y símbolos especiales, sin repetir patrones excesivamente simples, manteniendo diversidad y fluidez en la generación, y más allá de un simple muestreo aleatorio.

4 Riesgos y casos de uso

En las manos equivocadas cualquier herramienta puede ser potencialmente peligrosa. Esto también ocurre con PassGAN: las contraseñas que se generan tienen una alta probabilidad de ser correctas, por lo que los ciberdelincuentes pueden utilizarla para mejorar sus ataques.

En el caso de PassGAN, las contraseñas que se generan tienen una alta probabilidad de ser correctas. Los ciberdelincuentes pueden utilizar PassGAN para aumentar la eficacia de sus ataques. En lugar de confiar en listas de contraseñas comunes o predecibles, pueden utilizar PassGAN para generar contraseñas potenciales que son menos predecibles y, por lo tanto, más difíciles de defender.

Cabe destacar que solo cuando hay un filtrado de datos es cuando PassGAN es altamente peligrosa. Los ciberdelincuentes no obtienen acceso inmediato a los detalles de la contraseña en el momento en que un sitio web se ve comprometido; solo podrán acceder al "hash" encriptado de las contraseñas, que no es lo mismo que acceder a las contraseñas directamente. Además, tendrían que comprometer un servidor para acceder a las cuentas y violar la red de manera efectiva, lo que es muy difícil.

La eficacia de PassGAN en la generación de contraseñas realistas la convierte en una herramienta potencialmente peligrosa en manos de los ciberdelincuentes. Sin embargo, también destaca la importancia de implementar medidas de seguridad sólidas.

Para defenderse contra las amenazas que herramientas como PassGAN pueden presentar, es crucial implementar políticas de contraseñas fuertes, que incluyan la utilización de contraseñas largas y complejas, la rotación regular de contraseñas y el uso de autenticación de dos factores. También existen alternativas sin contraseña, como la biometría, pero no están libres de errores y sesgos. Además, las organizaciones deben cumplir con las regulaciones NIST y GDPR y escanear las vulnerabilidades frecuentemente.

5 Bibliografía

References

[1] Código de PassGAN. Disponible en: https://gist.github.com/PeterStaev/e707c22307537faeca7bb0893fdc18b7. Accedido en noviembre de 2025.

[2] Melicher, W., Ur, B., Segreti, S. M., Komanduri, S., Bauer, L., Christin, N., Cranor, L. F. (2017). *PassGAN: A Deep Learning Approach for Password Guessing*. Disponible en: https://arxiv.org/pdf/1709.00440. *arXiv preprint arXiv:1709.00440*.

[3] Lazarus. (2024). *PassGAN: Qué es y cómo afecta a la ciberseguridad*. Disponible en: https://www.lazarus.com.ve/passgan-que-es-y-como-afecta-a-la-ciberseguridad/. Accedido en noviembre de 2025.

[4] ManageEngine Argentina. (2024). *Los peligros de la IA generativa: la fortificación de contraseñas es el camino a seguir*. Disponible en: https://manageengine-argentina.prezly.com/los-peligros-de-la-ia-generativa-la-fortificacion-de-contrasenas-es-el-camino-a-seguir. Accedido en noviembre de 2025.