# ALGORITMO GENÉTICO PARA CIFRADO DE DATOS, BASADO EN UN NUEVO CONCEPTO PSEUDO-HEXADECIMAL CON INTELIGENCIA ARTIFICIAL

*Edgar Rangel Lugo[1], Kevin Uriel Rangel Ríos[1] , José Medrano Ochoa[1]*
*Carlos Alberto Bernal Beltrán[1] , Leonel González Vidales[1]*

*[1] Tecnológico Nacional De México. Instituto Tecnológico De Cd. Altamirano.*
*{ erangel_lugo@hotmail.com , kgvppro@gmail.com, Blackjosem7@gmail.com,*
*carlosalberto.bb@cdaltamirano.tecnm.mx , leonel.gv@cdaltamirano.tecnm.mx }*

**Resumen**

En Inteligencia Artificial (IA), existen técnicas, metodologías y paradigmas basados en métodos aleatorios y métodos heurísticos, que son aplicables en diversas áreas de estudio, por ejemplo: algoritmos genéticos, robótica móvil, redes neuronales, aprendizaje automático, reconocimiento de patrones, criptografía (basada en IA), por mencionar algunas. En el área de seguridad informática, podemos encontrar procesos que utilizan IA, donde se aprecia que una de las tareas más relevantes en dicha disciplina, consiste en prevenir el robo de datos o información. Para resolver este tipo de problemas, se utilizan estrategias de ciberseguridad, cuya reciente actualización, recibe el nombre de ciber-resiliencia. En dichos campos de estudio, se encuentra ubicada la criptografía, que proporciona métodos, técnicas y herramientas para el cifrado de datos o información. Se entiende por cifrado, la ocultación de la información, mediante la traducción de un mensaje original convirtiéndolo en un tipo de lenguaje o código, utilizando un alfabeto para cifrado/descifrado, que solamente podrá ser capaz de entender el software especializado o persona autorizada. Sin embargo, a pesar de la existencia de una gran variedad de algoritmos de cifrado; en la práctica actual, se observa que los usuarios de Internet, siguen estando vulnerables a ataques por parte de los "ciberdelincuentes" (hackers); y, si a ello le agregamos que, algunos lenguajes de programación (como: C, C#, C++, PHP, Java, Python, Ruby, VB, entre otros), pueden utilizar paquetes, funciones, clases o librerías que permiten el cifrado/descifrado de datos, ello les facilita a estos hackers apoderarse de información privada e importante, no solamente de usuarios particulares, sino también, de grandes organizaciones. Es por ello, que en la actualidad, es recomendable proteger la información utilizando algoritmos nuevos, los cuales, no incluyan soporte los lenguajes de programación libres o comerciales. Dicha situación, retrasa o demora a los hackers apoderarse de nuestra información. Por tal motivo, en este trabajo, se presenta una nueva alternativa para el cifrado de datos. Se trata de tres propuestas basadas en la primer versión estándar del conocido: Algoritmo de Caesar (Cifrado del César); siendo diseñadas utilizando números aleatorios para generar un alfabeto de cifrado/descifrado. Por lo tanto, la primera y segunda propuesta (Random Caesar I y Random Caesar II) trabajan la misma lógica computacional, pero difieren en la extensión del alfabeto de cifrado. La tercera propuesta presentada (Noised Pseudo-Hexadecimal GAs), fue planteado su diseño, usando un algoritmo genético con IA (denominado en esta investigación como: Noised Cyphered GAs Dictionary), basado en un modelo aleatorio, para generar el alfabeto de cifrado/descifrado, introduciendo un nuevo concepto, que en esta investigación, se ha denominado: Psedo-Hexadecimal, que consiste en introducir ruido a los números hexadecimales, para conseguir que el texto cifrado sea descifrado con mayor dificultad. Por último, cabe aclarar que el presente trabajo es un **reporte técnico** de la investigación, debido a que, al momento solamente se hace la presentación de los algoritmos y se muestran resultados preliminares, puesto que, falta concluir algunos experimentos que permitan demostrar o sugerir, que el cifrado/descifrado usando: Noised Pseudo-Hexadecimal GAs, permite generar mensajes seguros e indescifrables por las herramientas de software libre y/o comerciales que circulan en la actualidad.

**Palabras Clave**: *Inteligencia Artificial (IA), Algoritmos Genéticos (GAs), Criptografía, Cifrado/Descifrado de Datos, Algoritmo de Caesar (Cifrado del César), Números Hexadecimales, Métodos Aleatorios.*

## 1.- Introducción y Antecedentes

En el campo de las Ciencias Computacionales, existe una área llamada: Inteligencia Artificial (IA), cuyo propósito es hacer (emular o simular) que "la máquina piense" (Rangel, 2002). Para poder conseguir ese propósito, se apoya en varios métodos, técnicas, estrategias, modelos, arquitecturas y paradigmas; basados en métodos estadísticos (Kanal, 1963), métodos aleatorios (Reddaiah, 2019 ; Skalak, 1994 ; Kuncheva & Jain, 1999), métodos heurísticos, tales como: los basados en gráfos (Sánchez et. al. , 1997b) y basados en árboles (Ross-Quinlan, 1993), entre otros; que son aplicables en diversas áreas de estudio, por ejemplo: algoritmos genéticos (Kuncheva & Jain, 1999), robótica móvil (Rangel, 2002b; Rangel, 2004), redes neuronales (Bruzzone & Serpico, 1997), aprendizaje automático (Nils-Nilson, 1965), reconocimiento de patrones (Johns, 1961; Sebestyen, 1962; Rangel, 2002), criptografía (Delman, 2004; Mendoza, 2008; Kalsi et. al. , 2018; Reddaiah, 2019; Sebas, 2023) basada en IA, por mencionar algunas.

Se entiende por métodos heurísticos, a un conjunto de reglas definidas para resolver algo. Los gráfos y árboles de decisión, son un ejemplo de estructuras de datos, que utilizan este tipo de métodos (Sánchez et. al. , 1997b; Ross-Quinlan, 1993). En el presente trabajo de investigación, se utiliza este tipo de métodos, combinado con métodos aleatorios, para definir reglas y evitar que el alfabeto de cifrado/descifrado tenga valores repetidos.

Los métodos aleatorios, consisten en procesos basados en la generación de números tomados al azar, ya sea, con reemplazo (con repetición) o sin reemplazo (sin repetición). Un caso particular, son los algoritmos genéticos (Kuncheva & Jain, 1999) o GAs (por sus siglas del inglés: Genetic Algorithms), que adoptan como modelo computacional el comportamiento de la genética o evolución humana.

Según Reddaiah (2019), básicamente, los algoritmos genéticos estándar, inician con una población o conjunto de individuales cromosomas de determinado tamaño, que son formados directamente como entrada, de manera aleatoria e interactiva, definido por el usuario. La siguiente iterativa nueva población, que puede ser producida (reproducida) recibe el nombre de generación, y generalmente, es creada mediante un proceso de selección, siendo apoyado por un "wrapper" (clasificador) o una función de aptitud o "fitness" (por ejemplo: F=n+(€/m) ). Posteriormente, el proceso de selección, es una de las funciones, donde un par singular de cromosomas es escogido del grupo para efectos de reproducción de cromosomas. Ello se basa sobre el valor de la función de aptitud y puede ser implementado decidiendo de manera aleatoria, donde el cromosoma con "mayor aptitud" será considerado primero, para formar la siguiente generación, mediante un proceso de cruce. El cruce de cromosomas, es también, uno de los operadores genéticos, donde ello genera o produce nuevos hijos, al momento de cruzar dos cromosomas. Finalmente, se aplica la mutación, que permite a porcentaje de cromosomas sufrir alteraciones que podrían ser significativas para que exista variedad en la población de una o varias generaciones. En la presente investigación, se utiliza un GAs para asegurar que el alfabeto de cifrado/descifrado, esté bien revuelto, y produzca que una segunda ejecución del cifrado de datos, sea difícil que coincida con una anterior, aún cifrando el mismo texto, deberá cada ejecución tener un alfabeto ordenado diferente, aleatoriamente. Es decir, garantiza que un mismo texto, cifrado dos veces, tenga diferente alfabeto, y por ende, distinto contenido del paquete cifrado.

Los algoritmos genéticos, también son utilizados, por otro campo de estudio derivado de la IA, el cual, es conocido como: Reconocimiento de Patrones (Rangel 2002-2022) o Reconocimiento de Formas; donde destacan el uso de métodos para el Aprendizaje Supervisado (Rangel, 2002; Kuncheva & Jain, 1999), que permiten tomar decisiones, mediante la clasificación, predicción y evaluación de situaciones (patrones o formas) nuevas; así como, aplicar reducción o limpieza a volúmenes de información, utilizando bases de datos o muestras de entrenamiento supervisadas por un humano experto. Entonces, un método supervisado (Barandela R. & Juarez M. , 2001; Rangel, 2002), es aquel que aprende a partir de una muestra de entrenamiento (ME). Cabe mencionar, que todo método supervisado consta de dos etapas: Una etapa de aprendizaje y otra etapa de producción (Barandela R. & Juarez M. , 2001). Existen varios métodos o modelos supervisados, por ejemplo: árboles

de decisión como el caso del algoritmo C4.5 (Ross-Quinlan, 1993), método de Bayes y derivados (Johns, 1961), la Regla NN (por sus siglas en inglés: Nearest Neighbor), mejor conocida como: 1-NN o regla del vecino más cercano (Cover & Hart, 1967) y sus variantes (Rangel, 2002); así como, redes neuronales (Bruzzone & Serpico, 1997), aunque en este caso, también existen redes neuronales no supervisadas; por mencionar algunos modelos. Cabe mencionar que, en la etapa de aprendizaje, la mayoría de los métodos (excepto la Regla NN), suelen entrenar el modelo utilizando la muestra de entrenamiento (ME). Posteriormente, se desecha la ME y se procede a trabajar en la etapa de producción, donde comúnmente se utiliza para aspectos de clasificación que permiten apoyar en la toma de decisiones (Rangel, 2002). Por ejemplo, un árbol de decisión, en la etapa de aprendizaje crea el árbol y en la producción clasifica con dicha estructura. Las redes neuronales, en la etapa de aprendizaje, utiliza un modelo matemático para modificar los valores de sus enlaces, llamados "pesos", y en la etapa de producción, utiliza el mismo modelo matemático para clasificar nuevos patrones, usando los últimos valores de los pesos. En cambio, la regla del vecino más cercano, en la etapa de aprendizaje (Eui-Hong S. & Karypis G. , 1999; Ross Quinlan J. , 1993) consiste principalmente en el preprocesamiento de la muestra de entrenamiento. La elaboración o generación de la muestra de entrenamiento, requiere de un humano 'experto' del área en la que se va a trabajar para tener la seguridad de que la muestra de entrenamiento contiene patrones suficientes de cada una de las clases requeridas. En esta etapa de aprendizaje, se lleva a cabo el preprocesamiento de la muestra de entrenamiento, que se refiere a los métodos que se encargan de procesar la muestra de entrenamiento antes de llegar a la etapa de producción, para obtener un buen resultado en dicha etapa de producción. Dicho preprocesamiento consiste en eliminar 'ruido' de la muestra de entrenamiento, reducir la muestra de entrenamiento original generando una pequeña submuestra para aliviar un poco la carga computacional, entre otros aspectos. Pero, si consideramos que un método supervisado utiliza una muestra de entrenamiento, ya sea en la etapa de aprendizaje, y/o en la etapa de clasificación; entonces es necesario dar una definición formal de muestra de entrenamiento. Una definición formal de muestra de entrenamiento (Barandela, R. & Najera T. , 2002) podría ser la siguiente: ME = { (X1,Y1), (X2,Y2),....,(Xn, Yn) } ; es decir, aquel conjunto de los n pares (Xi, Yi), para i = 1,2, ...,n; donde la etiqueta 'Y' puede tomar valores en el conjunto {1, 2,..., m}, por lo tanto, 'Yi' designa la verdadera clase del patrón de entrenamiento 'Xi', entre las 'm' posibles clases, W1, W2, ... , Wm (Rangel E. , 2002). Entonces, una muestra de entrenamiento es creada por un experto humano, por tal motivo se considera que una muestra de entrenamiento es un conjunto de casos resueltos por el experto humano. Esos casos deben tener la identificación de clase (o la etiqueta) dada por el experto. Y debe haber casos de todas las clases que interesan. Un humano 'experto' es aquel usuario o individuo, que conoce bien determinada área (ambiente), en la que va a trabajar el sistema de reconocimiento de patrones (Rangel E. , 2002). En términos generales, los cromosomas de un algoritmo genético, pueden ser instanciados usando muestras de entrenamiento. En la presente investigación, solo se utiliza la Regla 1-NN como "wrapper" en el algoritmo genético; y se utilizan muestras de entrenamiento para evaluar los resultados preliminares de la experimentación, de la forma que se describe más adelante.

Por otra parte, un sistema de reconocimiento de patrones (supervisado), es aquel cuya función básica, consiste en detectar y extraer rasgos comunes a partir de los que describen los objetos de una misma clase, así como reconocer dicho patrón en cualquier situación nueva para clasificarlo o asignarlo en una de las clases que se consideran. Un objeto tiene características las cuales son llamadas también atributos. Un patrón 'Nuevo' es un objeto que tiene atributos pero no etiqueta. Un patrón de entrenamiento es un objeto, el cual, además de tener atributos, tiene una etiqueta que indica a que clase pertenece. Una clase de patrones, no se refiere solo a la etiqueta que tiene un grupo de patrones, sino más bien al grupo de patrones que tienen ciertas propiedades comunes (Rangel E. , 2002). En la presente investigación se utiliza un sistema de reconocimiento de patrones como "wrapper" (o clasificador 1-NN) para evaluar la población y seleccionar los cromosomas más aptos que formarán la siguiente generación del alfabeto de cifrado.

La regla del vecino más cercano (o Regla 1-NN) consiste primero en almacenar la muestra de entrenamiento en la memoria de la computadora, para poder encontrar aquel patrón de entrenamiento que se encuentra más cercano al nuevo patrón que se va a clasificar. Para lograrlo, se calculan las distancias entre el patrón nuevo y cada uno de los patrones que se encuentran en la muestra de entrenamiento. El patrón de entrenamiento que tenga la menor distancia, se dice que es el vecino más cercano al patrón nuevo. Entonces se asigna el patrón nuevo a la clase a la que pertenece su vecino más cercano. Para

calcular la distancia, se puede utilizar cualquier tipo de función de distancia. Las más utilizadas para datos numéricos son: la distancia de Manhattan (Kittipong Chomboon, et. al. 2015 ; Gongde Guo, et. al , 2004; Rangel, 2002; Wilson & Martinez, 1998) y la distancia Euclidiana (Rangel, 2002; Wilson & Martinez, 1998). Para datos nominales ó categóricos y binarios se utiliza la distancia de Gower o Heterogénea (Lewis & Catlett, 1994; Rangel E. , 2002). Dicha Regla 1-NN demostró tener dos principales deficiencias: Primero, dado que es una regla que depende de la muestra de entrenamiento, si dicha muestra de entrenamiento es muy grande, el algoritmo o regla de decisión es muy lento en ejecución. Segundo, debido a que el algoritmo depende de la memoria de la computadora, si la muestra de entrenamiento es muy grande tal vez la memoria de la computadora no sea suficiente para poder almacenar la muestra de entrenamiento. A pesar de las deficiencias de la regla 1-NN, se sigue utilizando debido a que garantiza una probabilidad de error baja (Rangel E. , 2002; Cover T. M. & Hart P. E. , 1967).

Es importante mencionar, que existen varios trabajos donde la Regla 1-NN (Cover T. M. & Hart P. E. , 1967) ha mostrado buenos resultados. Además, es un método supervisado que tiene muchas variantes. Una de las variantes mejor conocidas, es la regla de los k vecinos más cercanos, denominada también: “La Regla k-NN”. Esta Regla k-NN funciona de modo similar a la Regla 1-NN, la diferencia radica en que en la Regla 1-NN se busca solo el primer vecino más cercano (es decir, patrón cuya distancia es menor) y en la Regla k-NN se buscan los k vecinos más cercanos y se observa cual es la etiqueta mayoritaria, es decir, la etiqueta que predomina dentro de esos 'k-vecinos', y se asigna el patrón nuevo a la clase mayoritaria (Rangel E. , 2002). Existen varios estudios realizados con respecto a la Regla k-NN, uno de ellos fue propuesto por Hodges y Fix, quienes utilizaron la regla k-NN demostrando que tiene un buen desempeño en muestras de entrenamiento grandes (Fix E. & Hodges J. L. Jr. , 1952). En otro estudio destacado (Johns, M. V. , 1961), se utiliza k-NN en comparación con la Regla de Bayes. Aunque existen otros estudios más sobre k-NN (Kanal, L. N. , 1963; Sebestyen G. , 1962) donde sugieren el uso de NN en problemas de Reconocimiento de Patrones. Por su parte, Nils-Nilson (1965), también sugiere utilizar el vecino más cercano en problemas de Reconocimiento de Patrones y en áreas de aprendizaje de maquina (Machine Learning). En otro trabajo destacado, estudian el comportamiento de la regla k-NN para poder sugerir una pequeña modificación ó adaptación a dicha Regla k-NN (Loftsgaarden D.O. & Quesenberry C.P. , 1965). Por último, Cover & Hart demuestran que la Regla NN simple es admisible respecto a las Reglas: k-NN (con k distinto de uno) y Bayes; únicamente en distribuciones particulares. Donde Cover & Hart definen el término admisible aplicado a la comparación de reglas de decisión. Ellos consideran que una regla de decisión es admisible cuando no existe otra regla, entre las que se están comparando, mejor que ella (Cover T. M. & Hart P. E. , 1967). En la presente investigación, solamente se utiliza la Regla 1-NN sin variantes, aplicado en el algoritmo genético para generar el alfabeto de cifrado.

Sin embargo, en el área de Reconocimiento de Patrones, también incluye otra subárea de estudio conocida como: Minería de Datos (Hand et. al. , 2001) cuyo propósito es trabajar grandes volúmenes de información (mediante Datos Masivos, mejor conocida el área como: Big Data), y además de contribuir con métodos para el Aprendizaje Supervisado, también provee métodos y técnicas para trabajar el Aprendizaje No Supervisado, el cual, consiste en llevar a cabo, el agrupamiento de datos usado muestras, quedando clasificadas sin intervención del humano experto. Otro campo de estudio derivado, es el Aprendizaje Parcialmente Supervisado (Barandela & Najera, 2002), que consiste en trabajar con bases de datos o muestras de entrenamiento incompletas, cuyo propósito es el aprendizaje continuo (Barandela & Juárez, 2001) o automático, a través del reconocimiento de nuevas clases de patrones. En el presente trabajo de investigación, no se utiliza aprendizaje continuo, solamente se trabajó con Aprendizaje Supervisado, en la etapa de experimentación, clasificando los textos cifrados/descifrados, formando patrones dentro de una muestra de entrenamiento, cuya etiqueta de clase corresponde al mensaje o texto original. También, se utilizó Aprendizaje No Supervisado, en la parte del algoritmo genético, al momento de clasificar los cromosomas con un sistema de reconocimiento de patrones, para poder decidir cuáles fueron considerados más aptos y formar la siguiente generación (es decir, al clasificar con distancia euclidiana se tuvo que agrupar parcialmente, evaluando dos clases: de cromosomas más parecidos al alfabeto de una población inicial, que computaron la menor distancia y los cromosomas menos parecidos entre sí), que en términos generales, sobresalieron aquellos que no tenían caracteres ASCII repetidos después de la mutación; así como, aquellos que fueron observados con mayor distancia; es decir,

seleccionando lo contrario reportado por la Regla NN (Cover T. M. & Hart P. E. , 1967), reitero, menos parecidos entre sí, después de haber evaluado con la distancia euclidiana o inversa de la Regla 1-NN.

En el área de la Inteligencia Artificial (IA), particularmente, dentro del campo de los Métodos Supervisados, se entiende como deterioro de la clasificación de un sistema de Reconocimiento de Patrones (RP), cuando el modelo supervisado empleado comete demasiados errores durante la etapa o fase de producción. La disminución del rendimiento o decremento de la precisión de un sistema de RP, también conocida (en inglés) como: Global Accuracy (Lewis & Catlett, 1994) se ha observado en situaciones reales, tales como: en clases de plantas (iris), patrones obtenidos por señales de ultrasonidos (sonars), en reportes de resultados de análisis químicos constituyentes a encontrar tres tipos de vinos (wine), en problemas relacionados con rutas de vehículos (vehicle), en fallos de manufacturas de procesos, en estudios de clasificación de tipos de vidrios durante investigaciones de carácter criminologico que pueden ser usados como evidencia (glass), en raros diagnósticos médicos relacionados con diabetes tipo indian (pima) y registros clínicos de fallos cardíacos (heart), por mencionar algunos (Merz & Murphy, 1998 ; Frank & Asuncion, 2010). Por lo tanto, para medir la precisión, al clasificar patrones nuevos, es necesario establecer un criterio para conocer el porcentaje de acierto, y qué porcentaje de error fue el obtenido. Para medir el acierto se ha venido utilizando la Precisión del Sistema, conocida por su nombre en Inglés como: Global Accuracy. Un estudio realizado en este sentido, fue propuesto por Lewis y Catlett. Dichos autores realizan algunos estudios sobre la precisión global (Global Accuracy) en muestras de entrenamiento con varias clases. Los estudios realizados fueron respecto a muestras de entrenamiento con balance, es decir, no se trabajó el problema del desbalance de datos (Lewis D. & Catlett J. , 1994). Para calcular la Precisión (Global Acurracy) es necesario resolver lo siguiente: Precisión = ( (Total de Aciertos * 100) / (Total de Patrones Nuevos) ). En este caso, 'Total de Aciertos' es la suma de todos los aciertos obtenidos en la clasificación; y 'Total de Patrones Nuevos' es el número de patrones que se clasificaron (Rangel E. , 2002; Rangel, 2018-2022). En la presente investigación, se utilizó la Global Accuracy para medir la precisión de acierto o porcentaje de mensajes de textos cifrados y descifrados correctamente, con cada uno de los algoritmos propuestos en este trabajo.

Adicionalmente, es necesario hacer uso de métodos para estimación del error, ya que, dentro de los métodos supervisados, y en particular, la regla del vecino más cercano, durante la etapa de aprendizaje se lleva a cabo el preprocesamiento de la muestra de entrenamiento; y una vez finalizada dicha etapa, se procede a clasificar nuevos patrones. En dicha práctica, existen ocasiones donde es necesario tener una idea por adelantado acerca del porcentaje de error que se tendrá a la hora de clasificar estos patrones nuevos; todo con la finalidad de observar si se tiene un porcentaje de acierto elevado y aceptable, antes de pasar a la etapa de clasificación del método supervisado (Rangel E. , 2002). Existen varios métodos que pueden ser utilizados para la estimación de error, como el caso del Método L o 'Leave-One-Out' (Barandela R. & Gasca E. , 2000); el Método π (PI), conocido también como 'Validación Cruzada' (Rangel E. , 2002; Barandela R. , 2001), entre otros. En la presente investigación, solamente se ha utilizado la 'Validación Cruzada'. Dicho método, consiste en lo siguiente: (1) Se extrae un grupo de patrones de entrenamiento 'ME Pi' de tamaño 'Pi'. (2) El modelo se entrena con la muestra de entrenamiento ('ME') sin incluir a 'ME Pi'. (3) El modelo se evalúa con 'ME Pi'. (4) El proceso se repite para: i = 1, 2, ..., ( n / p ). Donde: 'p' es el porcentaje utilizado para 'ME Pi' que en ocasiones se le conoce como muestra de control ('MC'). En otras palabras, la 'Validación Cruzada' consiste en utilizar una parte de la muestra de entrenamiento como muestra de test ó de control. Lo más común es que se utilice el 10% ó 20% de la muestra de entrenamiento para formar parte de la muestra de control; el otro 80% ó 90% será utilizado para muestra de entrenamiento 'original o completa'. Cabe mencionar que en cada etapa o iteración, se va anotando el porcentaje de acierto obtenido, debido que al finalizar la última etapa, se calcula un promedio de todos los porcentajes de acierto obtenidos en cada fase. Dicho promedio, es la probabilidad de acierto estimada dada por la 'Validación Cruzada'. Por último, la precisión global, únicamente nos indica el porcentaje de acierto ocurrido durante la clasificación; pero no se puede observar si los resultados se desvían mucho ó no. Para poder medir, que tanto se desvían los resultados en cada una de las etapas de la validación cruzada; se calcula la Desviación Estándar del siguiente modo: S = RaízCuadrada ( ( Sumatoria (ACi - PROM)^2 ) / ( n - 1 ) ). Cabe aclarar que en la presente investigación, las muestras de entrenamiento contenían mensajes de texto cifrados (en valor numérico entero y hexadecimal correspondiente a caracteres de la tabla ASCII), y en lugar de hacer uso de la Regla 1-NN como clasificador, fueron sometidos al algoritmo de descifrado; y el resultado generado por el mismo, se comparó con

la etiqueta (que tenía el mensaje de texto original), marcando error si no coinciden y anotando acierto en caso de ser el mismo valor.

Con respecto a otros problemas que se pueden presentar en Reconocimiento de Patrones, dentro de los métodos supervisados, ha preocupado siempre el aspecto de la precisión en la clasificación. En problemas reales, los errores de clasificación pueden ser muy costosos, los cuales, pueden ser ocasionados por varios factores, entre los cuales podemos mencionar los siguientes: (a) Mala calidad en la muestra de entrenamiento; y (b) Elección del algoritmo de clasificación. La elección del algoritmo de clasificación no será estudiada, debido a que en la presente investigación, solamente se utiliza la regla del vecino más cercano, pero si cabe aclarar que la elección del algoritmo es un problema real, debido a que la regla 1-NN, k-NN y variantes, comparadas entre sí, difieren en resultados de clasificación, aunque en ocasiones para algunas muestras de entrenamiento tienden a obtener el mismo porcentaje de error (Rangel E. , 2002).

En lo que concierne con las muestras de entrenamiento de mala calidad, el primer factor a considerar es el siguiente (Rangel, 2002): Puede darse el caso de que el experto humano haya clasificado mal algunos patrones de entrenamiento, o también puede haber ocasiones que algún patrón de entrenamiento tenga uno o varios atributos que generen confusión, esto quiere decir, que dicho patrón tenga parecido (en cuanto a valores de los atributos) con elementos (patrones) de otra clase. Estos patrones de entrenamiento atípicos (o 'ruidosos') provocan confusión a la hora de la clasificación, y esto trae como consecuencia que se cometan errores. Un patrón atípico o 'ruidoso' se debe a que existe por lo menos un atributo que genera 'ruido', es decir, el patrón de entrenamiento contiene uno ó varios atributos que son confusos respecto a valores de atributos de otra clase, también un patrón se considera atípico si tiene una etiqueta errónea. Un segundo factor importante, es que en la etapa de aprendizaje, el humano 'experto' no haya considerado alguna clase, y por lo tanto, en la etapa de producción, al llegar un patrón nuevo de aquella clase que no fue contemplada se clasifica mal, siendo asignado a una clase que no pertenece (Rangel E., 2002). Otro factor es cuando existen muestras de entrenamiento desbalanceadas, es decir, que existen muchos más elementos de una clase que de otra, en una cantidad considerablemente grande (el doble, triple ó mayor). Las muestras de entrenamiento desbalanceadas (o 'No Balanceadas') se han considerado un problema muy serio, debido a que los errores de clasificación aumentan para la clase que tiene menor cantidad de patrones. Una clase mayoritaria es aquella clase que tiene el mayor número de elementos (patrones) que cualquier otra clase existente en una muestra de entrenamiento y una clase minoritaria, son aquellos elementos o el conjunto de patrones que existen en menor cantidad en la muestra de entrenamiento. Por último, otro factor a considerar, es cuando el experto no selecciona los atributos adecuados relacionados con el problemas y existen atributos que causan 'ruido' en la muestra de entrenamiento. Finalmente, otro factor importante es cuando una muestra de entrenamiento es muy pequeña; también se considera que una muestra de entrenamiento tiene mala calidad si existen más atributos que patrones por cada clase (Rangel E. , 2002). Entonces, una muestra de entrenamiento no balanceada es aquella en la que existen muchos más elementos (patrones ó representantes ) de una clase que de otra. Estas muestras de entrenamiento desbalanceadas (o 'No Balanceadas') se han considerado un problema muy serio, debido a que los errores de clasificación aumentan para la clase que tiene menos cantidad de patrones (Rangel E. , 2002). El problema de las muestras de entrenamiento 'No Balanceadas', fue detectado por primera vez en redes neuronales por Derouin et. al. (Kubat M. & Matwin S. , 1997 ]). Cabe mencionar, que en el presente trabajo de investigación, no se realizaron experimentos con ME desbalanceadas, y tampoco, se hizo la aplicación de ningún método de preprocesamiento de las muestras de entrenamiento. Sin embargo, sí se detectaron casos atípicos o ruidosos en los experimentos realizados con el algoritmo: Random Caesar I. Se descubrió que, al utilizar 255 caracteres del ASCII, hubo mucho error en el descifrado de datos, sobretodo en aquellos caracteres o valores ASCII que no son imprimibles en pantalla, tampoco fueron legibles en almacenamiento (archivos formato TXT) o en salida del campo de texto de la aplicación implementada en lenguaje Python3, lo ocasionó la reproducción de error.

Cabe mencionar, que dichas situaciones de error en muestras de entrenamiento que contienen almacenados patrones atípicos, han sido detectadas en otras investigaciones (Barandela & Gasca, 2000; Barandela et. al., 2021; Barandela et. al., 2003; Rangel, 2002), donde se ha observado, que tienden a producir significante deterioro de la clasificación con el modelo

estándar supervisado que ha sido seleccionado para trabajar en la etapa de producción. Se entiende como patrones atípicos, cuando un conjunto de registros (patrones) de particular clase (almacenados en la ME) contiene valores de sus atributos muy similares, comparados con los miembros de otra clase distinta, dentro de la misma ME. Podemos encontrar que este problema, es muy común en situaciones reales, antes mencionadas; y las propuestas de solución, suelen agruparse en dos categorías, principalmente: La primer propuesta, consiste en modificar el rendimiento del clasificador (modelo supervisado a emplear durante la etapa de producción), ya que, algunas veces, los errores de clasificación son producidos porque el método supervisado seleccionado puede ser inadecuado para trabajar con los datos de la ME o puede no ser muy eficiente para combatir el problema en cuestión. La segunda propuesta, consiste en aplicar pre-procesamiento a la ME original utilizando estrategias o métodos de Máquina de Aprendizaje, área también conocida (en inglés) como: Machine Learning (por ejemplo: uso de edición o aplicación de métodos para descontaminación de ME). Esto es, porque frecuentemente el deterioro de la clasificación se observa cuando existen patrones atípicos o clases desbalanceadas en la ME. Cabe aclarar, que en la presente investigación, no se incluyen resultados con ME desbalanceadas, por lo tanto, a la ME original no se ha aplicado ningún tipo de pre-procesamiento; pero sí se detectaron patrones atípicos, por ejemplo, una misma cadena de cifrado, generaba diferente resultado o mensaje de texto al ser descifrado.

Del mismo modo, comparando información presentada otros autores (Fix & Hodges, 1952), relacionado con el rendimiento de la regla del vecino más cercano (conocida como: 1-NN) en la clasificación de ME con patrones atípicos, se entiende que puede no ser expresado en términos de incrementar la precisión (global accuracy), en particular, cuando se utilizan ME grandes. En tales situaciones, la clasificación con el uso de una variante de la regla NN (1-NN), la conocida como regla de los k vecinos más cercanos (abreviado como: k-NN) es considerada una buena alternativa. Consecuentemente, en ambientes donde existe el problema de errores en la clasificación de patrones, algunas medidas han sido propuestas. Por lo tanto, la regla 1-NN usando criterio de distancia de manhattan (city-block), así como, el uso de la variante k-NN, podría ser considerado a futuro como un indicador para incrementar el rendimiento del clasificador (o modelo supervisado) o "wrapper" en el GAs que genera el alfabeto de cifrado, debido a que, las decisiones basadas en la regla k-NN, no dependen solamente de un ejemplar (más cercano) almacenado en la ME; y de este modo, surge la hipótesis de que podrían prevenirse algunas situaciones de cifrados atípicos.

Dentro de este contexto, existen aplicaciones recientes concernientes con respecto al uso de la Regla NN y sus variantes, que incluyen casos donde se utiliza la 'Distancia De Manhattan', entre otras métricas y modificaciones; donde que se comparan resultados con nuevas propuestas que aseguran mejoras al modelo tradicional. Además de los estudios mencionados previamente, para combatir problemas de reconocimiento de patrones; existen propuestas y aplicaciones en las cuales se ha venido utilizando la regla del vecino más cercano, en sus distintas variantes (Rangel E. , 2002). Un trabajo realizado, en el cual se utiliza la regla NN fue el propuesto por J.S. Sánchez y colaboradores. En dicho trabajo se refiere a la construcción de un grafo para 'Editar y Condensar' la muestra de entrenamiento haciendo uso de la regla NN (Sánchez J.S. et. al. , 1997). Tao Hong y colaboradores, diseñaron un algoritmo llamado: El más lejos vecino (algorithm k-nearest / farthest [FNN]). En dicho trabajo, proponen un algoritmo para reducción de búsqueda ó búsqueda veloz de patrones (Packing Prototypes and Fast NN Search). Dicho algoritmo de búsqueda rápida, basado en la regla del vecino más cercano (Fast NN Search Algorithm), y fue utilizado para el reconocimiento de texto ó caracteres japoneses (Hong T. et. al. , 1993). También, G. Tzanetakis y P. Cook, utilizaron la regla k-NN para clasificar archivos de sonido con formato mpeg (mp3); y utilizan un método llamado segmentación, el cual es aplicado a distintos tipos de música, entre ellos: rock con guitarra, piano instrumental, entre otros. La segmentación utilizada es con el fin de mejorar la calidad de sonido en archivos mpeg, así como mejorar la calidad de compresión en dichos archivos mpeg. Esta segmentación se refiere también al proceso de adaptación del audio en ciertas regiones que debería tener cierta textura ó calidad. El uso de la regla k-NN es utilizado después de haber realizado la segmentación, es decir, k-NN determina que archivo contiene mejor calidad de compresión y sonido en la hora de la clasificación (Tzanetakis G. & Perry C. , 2000). La regla k-NN también ha tenido su aplicación dentro de las bases de datos; un estudio realizado en este sentido, fue propuesto por Chakrabarti Kaushik, Kriengkrai Porkaew y Sharad Mehrotra. Dichos autores, proponen un método llamado: Reconstrucción Selectiva (selective reconstruction approach), donde utilizan la regla k-NN para aspectos de búsqueda y

clasificación en las bases de datos; también hacen uso de una variante de la regla k-NN, la cual es llamada: El multipunto k-NCN ó Multi-Distancia o MULTIDIS. Las bases de datos utilizadas, son bases de datos que almacenan objetos Multimedia, como lo son: documentos de texto, audio, video, imágenes, entre otros (Kaushik Ch. et. al. , 2000).

La regla k-NN ha sido utilizada también en otros aspectos de multimedia como lo son: los archivos de sonido; un estudio realizado para aspectos de archivos de sonido tipo MPEG-4 fue propuesto por: Khaled El-Maleh, Mark Klein, Grace Petrucci and Peter Kabal. Dichos autores utilizan la regla k-NN con valores para k=1 y k=3; donde utilizan las Reglas 1-NN y 3-NN para discriminación de archivos de sonido, es decir, para aspectos de clasificación utilizando diferentes archivos de sonido en formato MPEG-4. El tipo de música es de varias categorías o clases, entre ellas: música clásica, instrumental, opera, rock, dance, rap, y música pop. En el experimento se hace la comparación de técnicas. Las técnicas utilizadas para clasificar son: Regla de Bayes, 1-NN y 3-NN, entre otras. En dicho experimento se demuestra que la regla 1-NN y 3-NN obtienen mejores resultados (con Accuracy [no en estimación de Error]) en comparación con los resultados obtenidos con la regla de Bayes (Khaled El-Maleh et. al. , 2000). Por otra parte, Yin-Hung Kuo y Man-Hon Wong, comparan resultados obtenidos, en cuanto a clasificación y categorización automática de documentos vía Internet; dichos autores utilizan la regla k-NN para clasificar y llevar a cabo la categorización automática de documentos en Internet (Kuo Yin-Hung & Man-Hon Wong , 2000). Otros usos de variantes para la regla k-NN son aplicaciones para clasificación y aprendizaje, por ejemplo, en la categorización automática de texto (documentos), considerando el tremendo crecimiento en el volumen de documentos de texto disponible en Internet, bibliotecas digitales y otros medios; un trabajo relacionado, fue propuesto por Eui-Hong (Sam) and George Karypis, en el cual se utilizan variantes de NN para clasificar texto (Eui-Hong S. & Karypis G. , 1999); otro trabajo fue propuesto por Shrikanth Shankar y G. Karypis, donde utilizan la misma variante de NN para crear un algoritmo de ajuste de pesos (en una red neuronal) y poder asignar prioridades de clasificación de documentos o categorización automática del textos (Shrikanth S. & Karypis G. , 2000). Otro estudio con variantes de k-NN, fue realizado para la Clasificación de Niveles de Vértebras en un ser Humano, donde utilizan tres muestras de entrenamiento: una para LUMBAR, otra para CERVICAL, y una última para TORAXIC. En dicho experimento, se observa que la variante utilizada es muy lenta en ejecución al compararlo con el modelo tradicional de k-NN y k-Veloz NN (Fast k-NN), pero a pesar de tal deficiencia obtiene un porcentaje de error más bajo que k-NN y k-Veloz NN en la muestra CERVICAL (Moreno-Seco F. et. al. , 2001). En otro estudio relacionado, se observa el uso de la misma variante comentada, aplicado en 'aprendizaje' y clasificación, basados en criterios de vecindad, presentando métodos alternativos y análisis comparativos (Sánchez J. S. , 1998). En dichas aplicaciones, se ha demostrado que la regla k-NN y sus variantes, resultan ser un modelo muy eficaz (Rangel E. , 2002).

Dentro de este mismo contexto, existen otras áreas de estudio, que pueden trabajar modelos supervisados, mediante el uso de ME, por ejemplo: Aprendizaje Automático (conocida como: Machine Learning), y su sucesor: Aprendizaje Profundo (Deep Learning), que además, permite el estudio de Redes Neuronales Artificiales (RNA), con aprendizaje no supervisado (para agrupación de datos) y aprendizaje supervisado (para clasificar y predecir situaciones); que a saber, ambos, consisten en modelos matemáticos que permiten emular o simular la red neuronal del cerebro humano, pero, a pequeña escala.

Por otra parte, con la llegada de la IA, la mayoría de las áreas se actualizan a su versión 5.0. Por ejemplo, la Agricultura 4.0, Industria 4.0, Contabilidad 4.0 y otras áreas; consisten en el uso de maquinarias y tecnologías de precisión; mientras que en su actualización con 5.0, se entiende que ahora todo será automatizado, con Inteligencia Artificial, se realizarán las actividades de manera más automática.

El área de la Seguridad Informática, no fue la excepción. Ello quiere decir, que el campo de la ciberseguridad, ahora será capaz de responder de manera automática, y levantarse de ataques ocasionados por "ciberdelincuentes" (hackers o crackers), haciendo uso de IA. Este nuevo campo de actualización, recibe el nombre de ciber-resiliencia.

Dentro de las áreas de ciberseguridad y ciber-resiliencia, siempre ha preocupado el aspecto de mantener a salvo la información de los usuarios. Existen varias formas para conseguirlo, aunque una alternativa de estudio es conocida como Criptografía, la cual, también ya considera la IA dentro de sus metodologías; incluyendo entre sus estrategias, realizar el cifrado de los datos. En otras palabras, dentro del área de seguridad informática, podemos encontrar procesos que utilizan IA, donde

se aprecia que una de las tareas más relevantes en dicha disciplina, consiste en prevenir el robo de datos o información. Para resolver este tipo de problemas, se utilizan estrategias de ciberseguridad, cuya reciente actualización, recibe el nombre de ciber-resiliencia.

En el área de ciberseguridad (o ciber-resiliencia), existen varios trabajos al respecto, concernientes con el cifrado de datos. Un trabajo relacionado fue realizado por Rueda, et. al. (2005), quienes proponen una criptografía digital basada en tecnología óptica. La investigación presentada muestra resultados de la implementación de un algoritmo para cifrado digital de imágenes, cuyo sistema está basado en un arreglo de filtrado espacial usando la transformada de Fourier y una llave de cifrado de solo fase, para determinar los valores óptimos de distribución de la fase de la llave. Cabe mencionar, que desde la década de los noventa la tecnología óptica hace presencia en la construcción de arreglos de cifrado (Rueda; et. al, 2005). En 1994 Javidi y Horner proponen el método para encriptación óptica de información que usa mascaras aleatorios de fase. A este trabajo han seguido otros que usan llaves ópticas aleatorios de fase en el plano de entrada y en el plano de Fourier, sistemas de encriptación cuando existe ruido y distorsión, patrones de SPECKLE encriptados mediante máscaras de fase aleatorias y encriptación mediante conjugación de fase en un cristal fotorrefractivo (Javidi 1997; Tajahuerce 2000-2001; Rosen, 2001; Mogensen 2001-2000; Matoba 2000; Nomura 2000; Sinha 2003; Rueda, 2002), en entre otros; recientemente se presentó un método que usa la transformada wavelet (Linfei 2005). Por otra parte, si consideramos el dispositivo móvil como una herramienta para la transmisión de datos, los cuales son susceptibles a las amenazas en los canales de transmisión en la red. La seguridad informática cumple un papel muy importante para garantizar la disponibilidad, privacidad e integridad de la información (Solís ; et. al. , 2017).

En los campos de estudio relacionados con la ciberseguridad y ciber-resiliencia, se encuentra ubicada la criptografía, que proporciona métodos, técnicas y herramientas para el cifrado de datos o información. Se entiende por cifrado, la ocultación de la información, mediante la traducción de un mensaje original convirtiéndolo en un tipo de lenguaje o código, utilizando un alfabeto para cifrado/descifrado, que solamente podrá ser capaz de entender el software especializado o persona autorizada.

El uso de criptografía, cifrado y descifrado de texto, fue iniciado alrededor de 1900 B.C. ; cuando los Egipcios comenzaron a aplicar procedimientos de correspondencia (Hebert, s.f. ; Reddaiah, 2019). La criptografía ha sido usada casi simultáneamente desde el desarrollo avanzado del lenguaje escrito (Singh, 2000) y tradicionalmente jugó un rol fundamental en la protección de las comunicaciones oficiales de los Estados, de los gobernantes y, principalmente, de las instituciones militares (Álvarez, 2019). Fueron precisamente los grandes conflictos bélicos del siglo XX, los que permitieron el desarrollo de los principales avances en la criptografía moderna, especialmente a partir de la invención de los métodos de computación electrónica -- que facilitaron el procesamiento de grandes volúmenes de información -- y el desarrollo de un ámbito específico de las matemáticas: las teorías de la información de Claude Elwood Shannon, quien es considerado el padre de la criptografía moderna (Granados, 2006 ; Singh, 2000). Otra área donde la criptografía tiene impacto, es en las redes mundiales como la Internet y las redes telefónicas, a través de ellas se manejan grandes volúmenes de información confidencial: de tarjetas de crédito, de transacciones bancarias, corporativa y/o gubernamental, etc (Rueda; et. al, 2005). En América Latina existen antecedentes del uso de herramientas de cifrado de comunicaciones desde la Conquista y su uso se intensificó durante la Colonia y los procesos independentistas (Galende, 2000).

Sin embargo, la criptografía es una palabra que proviene de las palabras griegas Kriptos (ocultar) y graphos (escritura). Literalmente significa “escritura oculta”, es decir mantener seguro los secretos mediante la codificación de los mensajes para hacerlos ilegibles, de tal manera que solo pueda verla aquel receptor que el emisor desea. El proceso de convertir el texto claro en texto cifrado se denomina cifrado; y el proceso de recuperar el texto claro a partir del texto cifrado se denomina descifrado (Mendoza, 2008). El propósito de la criptografía es transmitir un mensaje entre emisor y receptor, de modo que, sea incomprensible. Para ello, no sólo es necesario un robusto algoritmo, sino también, una fuerte clave y robusto proceso para cifrado y descifrado de datos (Kalsi et. al. , 2018).

La implementación de sistemas de seguridad de la información, es un tema que cada día toma mayor importancia. Con la aparición de los ordenadores digitales, se inicia una amplia investigación, basada en la algoritmia matemática aplicada al

cifrado de información; estas técnicas aun se mantienen en desarrollo y mejoramiento, pero las mismas se hacen vulnerables en la medida que los procesadores digitales se hacen cada vez más robustos, con capacidad para resolver problemas matemáticos de alta complejidad (Rueda; et. al, 2005).

La criptografía no es sólo una ciencia que aplica a matemáticas complejas y lógica para diseño de robustos métodos que permiten la ocultación de datos, denominado como: encriptación (Kalsi et. al. , 2018). También, incluye la parte del descifrado de datos.

Según Solis et. al. (2017), una de las técnicas que ayuda en ésta tarea es la criptografía, cuyo fundamento es transformar un mensaje de modo que sea inentendible salvo para los que posean la clave para descifrarlo. Una investigación de dichos autores, se enfoca en el uso del algoritmo RSA entre dispositivos móviles, los datos cifrados se envían por canales de comunicación llamados hilos que mediante fórmulas y procesos ejecutados en el servidor, ayudarán a ejecutar el cifrado y descifrado de los datos (Solís ; et. al. , 2017). En este contexto, durante la investigación se diseñó un prototipo para el intercambio de datos entre dispositivos móviles de manera inalámbrica.

Los algoritmos criptográficos utilizados desde la época de la antigua Roma hasta nuestros días, son métodos que convierten un mensaje de texto plano en texto cifrado. El proceso inverso, se conoce como "descifrar", y consiste en llevar el texto cifrado a texto claro. Usualmente estos algoritmos utilizan una llave secreta como parte de la entrada (Gómez; et. al. , 2012). Un algoritmo criptográfico se caracteriza por convertir un texto claro, en otro, llamado texto cifrado. El contenido de la información es igual al anterior pero solo puede ser entendido por la persona autorizada (Van & Jajodia, 2011). Los canales de comunicación se suponen inseguros cuando se desea enviar información confidencial a través de estos, en cuyos casos se requiere cifrar el mensaje (Fulgueira; et. al , 2015).

En términos simplificados, el cifrado de comunicaciones consiste en la utilización de un algoritmo matemático que «envuelve» un mensaje de manera que solo el receptor legítimo pueda abrirlo y hacerse de su contenido, mediante la utilización de una llave o clave única que «desenvuelve» el mensaje. El propósito del cifrado es hacer ininteligible un mensaje a los ojos de un tercero ajeno a la comunicación (Álvarez, 2019); ya que, es necesario tener un modo seguro de transmisión de datos, desde un lugar a otro; a través de diferentes mecanismos y/o servicios de seguridad, que son necesarios para proporcionar integridad y seguridad a los recursos propios. Los mecanismos que proporcionan seguridad, deben estar actualizados constantemente para prevenir ataques. En este proceso, deben incluirse funciones especiales robustas (Reddaiah, 2019).

Los algoritmos de cifrado se clasifican en simétricos y asimétricos. La criptografía simétrica utiliza la misma clave para cifrar y descifrar el mensaje de datos, es decir se basa en un secreto compartido (Gómez; et. al. , 2012). Los métodos conocidos como cifrado de llave simétrica, la llave debe coincidir tanto en el proceso de cifrado como en el de descifrado para que la comunicación entre el emisor quien cifra, como el receptor que descifra, sea exitosa. El objetivo de realizar el cifrado, es dificultar (o imposibilitar) la comprensión del mensaje a otra persona distinta del receptor legitimo del mensaje. Idealmente los emisores y receptores legítimos del mensaje deben ser los únicos que conozcan la llave secreta, ya que es en esta llave secreta que radica la privacidad de la comunicación (Gómez; et. al. , 2012). En otras palabras, los algoritmos asimétricos, en lugar de usar una sola clave para realizar la codificación y la decodificación, se utilizan dos claves diferentes: una para cifrar y otra para descifrar. Estas dos claves se encuentran asociadas matemáticamente, cuya característica fundamental es que una clave no puede descifrar lo que cifra (Mendoza, 2008). Entonces, un algoritmo simétrico usa una sola clave para cifrar/descifrar; a diferencia del asimétrico, que usa dos claves: una pública y otra privada. En el presente trabajo de investigación, solamente se utilizan algoritmos simétricos.

Una investigación similar, fue presentada por Mendoza (2008), donde se incluye una breve introducción a la criptografía, sin profundizar en las matemáticas que soportan los algoritmos criptográficos; mostrando una visión de los distintos tipos: simétricos y asimétricos; aplicando ambas técnicas mediante el uso de los algoritmos: RSA y 3DES, implementados en Visual Basic. NET (Mendoza, 2008). Del mismo modo, Jiménez; et. al. (2015), proponen un algoritmo de cifrado simétrico que toma como entrada la información original de longitud L y al codificarla genera el texto cifrado de longitud mayor LM; implementando

el sistema discreto caótico mapa logístico para generar tres órbitas diferentes: la primera se utiliza para aplicar una técnica de difusión con la finalidad de mezclar la información original, la segunda órbita se combina con la información mezclada y se incrementa la longitud de L hasta LM y con la tercer órbita se implementa la técnica de confusión. El algoritmo de cifrado se aplicó para codificar una imagen. Por su parte, Fulgueira; et. al. (2015) presentan un proceso de paralelización del algoritmo criptográfico GOST. Dicho trabajo se enfoca en la reducción del tiempo de ejecución de la implementación del algoritmo criptográfico GOST (Estándar Gubernamental de la Unión de Repúblicas Socialistas Soviéticas). Según Fulgueira et. al. (2015) los detalles del algoritmo fueron publicados en 1990 bajo el nombre de Estándar Soviético (GOST 28147-89). El estándar provee un nivel de seguridad flexible y puede ser empleado para proteger información en sistemas de cómputo y en redes de computadoras (Courtois, 2012; Pieprzyk & Tombak,1994). El estudio no se encuentra enfocado al análisis de fortaleza del algoritmo criptográfico; solo se realiza un diseño paralelo basado en la metodología de Lan Foster, el cual es aplicado a dos implementaciones usando técnicas como: OpenMP y CUDA, logrando el mejor resultado empleando este último. Otro trabajo relacionado, fue presentado por: Álvarez (2019), quien expone algunos aspectos jurídicos del cifrado de comunicaciones. En dicho trabajo, explora algunas de las consideraciones jurídicas más relevantes sobre la relación entre cifrado de comunicaciones y ciertos derechos constitucionales, con enfoque en el derecho chileno, sosteniendo que existe una tensión no resuelta en el rol que ejerce el uso de herramientas de cifrado de comunicaciones respecto del ejercicio de esos derechos fundamentales y la seguridad pública.

Por otra parte, se conoce como criptoanalista a la persona que sin tener la llave secreta, trata de descifrar un texto encriptado. Los estándares criptográficos actuales exigen que aunque el algoritmo de encripción sea conocido por el criptoanalista, si este no posee la llave secreta, no sea factible que este consiga obtener todo o parte del texto plano. Es por este motivo que al diseñar algoritmos criptográficos modernos se deben conocer los métodos de criptoanálisis (Gómez; et. al. , 2012). El conocimiento de métodos de criptoanálisis permite al diseñador de algoritmos criptográficos analizar qué debilidades y fortalezas tienen los algoritmos, y de esta forma estar mejorándolos continuamente (Menezes; et. al. , 1997 ; Gómez; et. al. , 2012).

En lo que concierne con los algoritmos de cifrado; en el pasado, Julio César (Julius Caesar), también diseñó métodos seguros para transmitir en secreto información para gente importante en el campo de la milicia (William, 1999 ; Reddaiah, 2019). La seguridad proporcionada por cripto-sistemas se basa completamente en la construcción de mecanismos con una gran variedad de funciones y operaciones (Reddaiah, 2016).

Existen varios métodos o algoritmos para el cifrado de datos (Barranco & Galindo, 2022): por desplazamiento, por sustitución, por permutación, cifrados de flujo, de claves privadas (como: DES, AES), de clave pública (como: RSA), basados en funciones Hash (como: MD5, SHA1 y derivados), entre otros.

Uno de los algoritmos más populares en sus inicios, por ser sencillo de implementar, fue el algoritmo de desplazamiento o sustitución, conocido como: Algoritmo Caesar o Cifrado del César (Barranco & Galindo, 2022; Gómez et. al. , 2012); el cual, tiene algunas variantes o derivados, que han desarrollado distintos investigadores. Al respecto, existe un trabajo relacionado, presentado por Gómez; et. al. (2012), donde se muestran algunas técnicas de encriptación clásicas, tal como los cifrados del Cesar y Vigenère; realizando un cripto-análisis sobre métodos clásicos de cifrado. Dicha publicación, muestra algunas técnicas básicas y modernas de cripto-análisis basadas en la teoría de la información y la estadística, argumentando que pueden ser usadas en otros métodos de cifrado si las mismas debilidades están presentes, por esta razón, según Gómez (2012), es útil conocer estas técnicas de cripto-análisis cuando se diseñan nuevos algoritmos. En la presente investigación, se utiliza el Cifrado del Caesar estándar, realizando una modificación en la ecuación para cifrado de datos, entre otras modificaciones, que serán referidas más adelante.

Empero, si observamos en la literatura, podemos encontrar que, los cifrados basados en Caesar, ya no son muy utilizados recientemente; porque destacan los siguientes: MD5, SHA1 y derivados basados en tablas o funciones Hash; así como, otros muy populares son: AES, DES, RSA, por mencionar algunos (Progress Software Corporation, et. al. , 2020-2022).

Lo anterior, puede ser debido a que, los fraudes y ataques a los sistemas informáticos son cada vez de mayor preocupación, siempre que las pérdidas por tales prácticas ilícitas implican incalculables pérdidas económicas, tanto a usuarios de sistemas financieros como a la banca misma, a estamentos gubernamentales, y en general a todas las empresas que manejan bases de datos a través de la Internet o cualquier otro medio de transporte de la misma (Rueda; et. al, 2005). Según Jiménez et. al. (2015), es indispensable utilizar algún mecanismo que ayude a resguardar la información de algún ataque malicioso, uno de los más utilizados es la criptografía que se encarga de escribir en secreto, proporcionando confidencialidad a la información mediante un método de cifrado (Oppliger, 2005). Muchos métodos o esquemas de comunicación segura se han desarrollado para cifrar información basándose en sistemas discretos caóticos (Hossam et al., 2007; Pisarchik y Zanin, 2008; Pisarchik y Flores, 2006; Ranjan y Saumitr, 2006). Esto se debe a la relación cercana que existe entre el caos y la criptografía; porque los sistemas caóticos tienen características como: ergodicidad, propiedades de mezcla, sensibilidad a los parámetros y las condiciones iniciales, que pueden considerarse análogos a las técnicas de difusión y confusión integrados en muchos sistemas criptográficos (Chong et al., 2011; Jiménez et. al. , 2015).

Además de los sistemas caóticos discretos y las variantes de Julio César, entre otros algoritmos, también existe la tecnología óptica. Según refiere Rueda et. al. (2005), desde la década de los noventa la tecnología óptica hace presencia en la construcción de arreglos de cifrado. En 1994 Javidi y Horner proponen el método para encriptación óptica de información que usa mascaras aleatorios de fase. A este trabajo han seguido otros que usan llaves ópticas aleatorios de fase en el plano de entrada y en el plano de Fourier, sistemas de encriptación cuando existe ruido y distorsión, patrones de SPECKLE encriptados mediante máscaras de fase aleatorias y encriptación mediante conjugación de fase en un cristal fotorrefractivo (Javidi 1997; Tajahuerce 2000-2001; Rosen 2001; Mogensen 2001-2000; Matoba 2000; Nomura 2000; Sinha 2003; Rueda,2002), en entre otros; recientemente se presentó un método que usa la transformada wavelet (Linfei 2005).

A pesar de la existencia de demasiados algoritmos de cifrado, como son: RSA, MD5, DES, AES, SHA1 y familia de algoritmos derivados, entre otros; se han visto vulnerables a ataques en la red. Por ejemplo, se han registrado muchos ataques contra el algoritmo DES, que lo han convertido en un método inseguro de cifrado (Rodríguez, 2020). Sin embargo, a pesar que AES, ha sido considerado un algoritmo seguro, debido a que, el único tipo de ataque efectivo contra AES ha sido el ataque de fuerza bruta, y ello hace que AES se siga considerando como un esquema de cifrado seguro (Rodríguez, 2020); se estima que, en el futuro, con el desarrollo de los computadores y el procesamiento en red, se podrá romper el RSA, particularmente con la llegada de la computación cuántica (Thakur & Kumar, 2011).

Sin embargo, a pesar de la existencia de una gran variedad de algoritmos de cifrado "seguros"; en la práctica actual, se observa que los usuarios de Internet, siguen estando vulnerables a ataques por parte de los “ciberdelincuentes” (hackers); y, si a ello le agregamos que, algunos lenguajes de programación (como: C, C#, C++, PHP, Java, Python, Ruby, VB, entre otros), pueden utilizar paquetes, funciones, clases o librerías que permiten el cifrado/descifrado de datos, ello les facilita a estos hackers apoderarse de información privada e importante, no solamente de usuarios particulares, sino también, de grandes organizaciones. Esto es, desafortunadamente, porque el proceso de cifrado/descifrado de los algoritmos populares es conocido por los desarrolladores; y si a ello le agregamos, lo antes mencionado, con respecto a que varios lenguajes de programación (como: C, C#, C++, PHP, Java, Python, Ruby, VB, entre otros), ya incluyen o pueden utilizar paquetes, funciones, clases o librerías que permiten el cifrado/descifrado de datos de los algoritmos más populares; ello hace vulnerable la seguridad de los datos e información de los usuarios, no solamente en Internet, sino también, dentro de un contexto local.

Lo anterior, debido a que, si determinado lenguaje de programación, permite el cifrado, por ejemplo, del algoritmo MD5; solo bastaría "entrenar un modelo supervisado" durante un lapso largo de tiempo, para construir un diccionario con IA; ya sea, mediante procesamiento de lenguaje natural y/o métodos basados en semántica web, para poder descifrar, por ejemplo, un texto o clave de 16 caracteres.

Es por ello, que se considera importante diseñar nuevos algoritmos de cifrado, o en su defecto, realizar variantes o modificaciones, a los algoritmos ya existentes, para que los "ciber-delincuentes" desconozcan el procedimiento de descifrado, y al menos, tener segura nuestra información, durante un breve lapso de tiempo, mientras logran descifrar el modelo

matemático empleado. Por lo tanto, es recomendable proteger la información utilizando algoritmos nuevos, los cuales, no incluyan soporte los lenguajes de programación libres o comerciales. Dicha situación, retrasa o demora a los hackers apoderarse de nuestra información. Por tal motivo, en este trabajo de investigación, se presentan tres alternativas "inteligentes", para el cifrado de datos, una de ellas, llevada a cabo, mediante IA, a través de un algoritmo genético.

La alternativa de cifrado de datos utilizando algoritmos genéticos (GAs), no es algo nuevo; debido a que, ya se han utilizado en otras investigaciones (Reddaiah, 2019; Tanenbaum, 2003; Delman, 2004; Kalsi et. al. , 2018; Rodríguez, 2020). Además, los algoritmos genéticos (Reddaiah, 2019 ; Skalak, 1994 ; Kuncheva & Jain, 1999), se han aplicado con éxito en una variedad de problemas, desde la optimización de rutas de transporte hasta la planificación de redes de energía y la detección de fraudes en el sector financiero, robótica, diseño de circuitos electrónicos, aprendizaje automático (Sebas, 2023); en criptografía basada en Inteligencia Artificial, en el campo del comercio electrónico (Reddaiah, 2019), por mencionar algunas áreas. El comercio electrónico (e-commerce) es un campo que esta creciendo rápidamente, observando medidas de seguridad en los datos del negocio e información de los clientes (Reddaiah, 2019). Podemos observar en la literatura, un atractivo y extenso trabajo al respecto (Khan, 1967); así como, una gran variedad de elementos de sistemas modernos desarrollados (Tanenbaum, 2003).

Tal como ya se ha descrito previamente, un algoritmo genético, es un clase de optimización de algoritmos que puede resolver problemas modelando una versión simplificada del proceso genético humano (Delman, 2004). Según Reddaiah (2019), básicamente, los algoritmos genéticos estándar (Reddaiah, 2019 ; Skalak, 1994 ; Kuncheva & Jain, 1999), inician con una población o conjunto de individuales cromosomas de determinado tipo y tamaño, que son formados directamente como entrada, de manera aleatoria e interactiva (Reddaiah, 2019), definido por el usuario. Se entiende por cromosoma, a un patrón o vector de tipo bit o binario (0s y 1s), aunque el tipo de datos puede ser cualquiera, por ejemplo: booleano (true/false), caracter (char), entero (byte, int, long), real (float, double) u otro tipo. La siguiente iterativa nueva población, que puede ser producida (reproducida) recibe el nombre de generación, y generalmente, es creada mediante un proceso de selección (Reddaiah, 2019), siendo apoyado por un "wrapper" (clasificador) o una función de aptitud o "Fitness Function" (por ejemplo: F=n+(€/m) ). La selección es una de las funciones, donde un singular cromosoma es escogido del grupo para efectos de reproducción de cromosomas. Ello se basa sobre el valor de la función de aptitud y puede ser implementado decidiendo de manera aleatoria, donde el cromosoma con "mayor aptitud" será considerado primero, para formar la siguiente generación, mediante un proceso de cruce. El cruce de cromosomas, es también, uno de los operadores genéticos, donde ello genera o produce nuevos hijos, al momento de cruzar dos cromosomas. Por ejemplo, algunos atributos del primero y segundo cromosoma se mezclan para formar un tercero, que será evaluado con la función de aptitud para decidir si forma parte de la siguiente generación. El cruce puede ser de tres tipos: (1) Usando un simple punto, partiendo el cromosoma a la mitad o en un extremo seleccionado aleatoriamente; (2) Usando dos puntos, seleccionando dos límites para mezclar (intercambiar) solo los bits centrales, o si se prefiere, se puede intercambiar o cruzar por los extremos del cromosoma; y, (3) Cruce uniforme, donde se toma un par de bits de cada cromosoma padre y madre, para mezclar (cruzar) de manera uniforme hasta finalizar el cromosoma (Reddaiah, 2019). Otra función de los GAs es la mutación, donde al menos uno de los bits del cromosoma (o cierto porcentaje) es modificado usando una función de mutación. Este tipo de función, puede ser de dis maneras: (1) Mutación Flipping: Donde uno o más bits son cambiados (mutados) de 0 a 1, o de 1 a 0, según el valor que tenga en el cromosoma; (2) Mutación Boundary: Por límite o frontera, donde se realiza el intercambio del bit de un extremo (límite) a otro previamente seleccionado. Después de aplicar: selección, cruce y mutación; los cromosomas de la población, en la actual generación, serán evaluados (con función de aptitud o "wrapper") para descartar aquellos con la menor aptitud y repetir el proceso: selección-cruce-mutación, en la población de la siguiente generación. Por último, en lo que concierne con la función de aptitud (Fitness Function) de los GAs, según Reddaiah (2019), está basada sobre una matemática ecuación, o al menos, son las comúnmente más usadas. Una buena función de aptitud es útil para explorar en la búsqueda eficientemente, mediante el uso de valores matemáticos, donde el valor de cada aptitud individual (del cromosoma) es evaluado sobre la base de símbolos que se repiten más número de veces. Un ejemplo de este tipo de función (de aptitud), puede ser (Reddaiah, 2019): F=n+(€/m). Donde: F = Funcion de aptitud (Fitness function); "n" es el número total de símbolos usados en la llave de formación (es decir, "genes" o bits en el cromosoma); "m" es el porcentaje de máxima aparición de los símbolos (por ejemplo: 50%-50% o 70%-30%, por mencionar algunos); "€" es el ideal

Reserva de Derechos al Uso Exclusivo No. 04-2023-091910490900-102, ISSN: En trámite. Año 1, No. 2, Marzo – Mayo 2024.
Fecha de Recepción: 03/12/2023| Fecha de Aceptación: 11/12/2023

porcentaje de cada símbolo (es decir, el porcentaje esperado); por lo tanto, existen diferentes tipos de operadores que son frecuentemente usados en GAs para obtener seguridad y resultados con precisión. Además de la función de aptitud, antes mencionada, se puede usar un "wrapper" (por ejemplo, un clasificador) para evaluar los cromosomas más aptos de cada generación. En Kuncheva & Jail (1999); así como, en Mitchell & Holland (1993), se puede observar que utilizan GAs evaluando con la regla del vecino más cercano (Cover & Hart, 1967; Hart, 1968; Rangel, 2002; Kittipong-Chomboon, 2015 ; Rangel, 2019-2022; Rangel & Rodríguez, 2018; Gongde-Guo et. al. , 2004; Wang, 2002; Barandela et. al. , 2001a ; Hong-Tao et. al. , 1993; Loftsgaarden & Quesenberry, 1965; Fix & Hodges, 1952; Dudani, 1976; Lewis & Catlett, 1994); y se hace con la finalidad de reducir dimensión en muestras de entrenamiento, comparando resultados con Skalak (1994), quien no hace uso de GAs, pero reduce la dimensión de manera sorprendente, aunque ello, afecte la calidad de precisión. En las dos primeras investigaciones, se lleva a cabo la evaluación de la aptitud, usando algún método de estimación de error (Rangel, 2019-2022; Barandela & Gasca, 2000; Rangel, 2002; Barandela & Juarez, 2001; Barandela et. al., 2001); lo anterior, es utilizado para poder decidir cuáles cromosomas de cada generación, resultan ser los "más aptos". Cabe mencionar que en dichas investigaciones, no se trabaja el cifrado de datos, solamente se utilizan GAs con el propósito de reducir la dimensión o tamaño de las muestras (Ritter et. al. , 1975; Barandela, 2001; Hart,1968; Tomek,1976; Rangel, 2002; Toussaint, 1992; Gates, 1972; Wilson & Martinez, 1998; Kubat et. al. , 1997; Kubat & Matwin, 1997; Sánchez et. al. , 1997b ; Aha, 1991; Swonger, 1972), o en su defecto, limpiar patrones atípicos o ruidosos (Wilson, 1972; Barandela & Gasca, 2000; Koplowitz & Brown, 1978; Rangel, 2002; Barandela & Rangel et. al. 2003; Dasarathy et. al. , 2000; Sánchez et. al. , 2001 ; Wilson & Martinez, 2000 ; Sánchez et. al. , 1997a ; Wilson & Martinez, 1997 ; Devijver & Kittler, 1980; Tomek, 1976b), empleando en algunos experimentos, el uso de muestras de entrenamiento desbalanceadas (Barandela, 1990; Rangel, 2019; Barandela & Rangel et. al. 2003; Rangel, 2002), muestras que en su mayoría, fueron tomadas de repositorios (UCI: Frank & Asuncion, 2010 ; ELENA DATABASE, 2002; UCI: Merz & Murphy, 1998).

En el contexto del cifrado de datos usando GAs, podemos observar en la literatura varios trabajos al respecto, ya que, según Sebas (2023), los algoritmos genéticos se han aplicado con éxito en una variedad de problemas, desde la optimización de rutas de transporte hasta la planificación de redes de energía y la detección de fraudes en el sector financiero, robótica, diseño de circuitos electrónicos y aprendizaje automático.

Existe una investigación desarrollada por: Delman (2004), quien explora el uso de algoritmos genéticos (GAs) en criptografía. Se trata de dos propuestas, una concerniente con cripto-análisis y otra utilizando una metodología basada en GAs. Ambos, fueron implementados en un software. Los resultados comparados revelan retardo de tiempos y porcentaje de desencripción satisfactoria. Del mismo modo, Kalsi et. al. (2018), presentan una investigación concerniente con cifrado de datos usando GAs, introduciendo un nuevo concepto de ADN Criptográfico con Aprendizaje Profundo (DNA Deep Learning Cryptography), que está definido por una técnica de ocultación de datos en los términos de una secuencia de ADN (médico) para aprendizaje automático, diseñado del siguiente modo: En la técnica criptográfica, cada alfabeto de una letra es convertido en una diferente combinación de cuatro bases, denominadas: Adenina (Adenine, A), Citosina (Cytosine, C), Guanina (Guanine, G) y Thymina (Thymine, T); los cuales, construyen el ácido deoxy-ribonucleico humano, conocido como: ADN (DNA). Para construir el ADN computacional en un nivel digital, son propuestos fáciles y efectivos algoritmos; implementado primero, un método para generación de la llave o clave, basado en la teoría de selección natural de los GAs con NW (Needleman-Wunsch) estrategias. Posteriormente, un método secundario se ha implementado para la encriptación y desencripción, basado en ADN computacional, usando "biológicas operaciones", como son (Kalsi et. al. , 2018): Transcripción (Transcription), Translación (Translation), Secuencia de ADN (DNA Sequencing) y Aprendizaje Profundo (Deep Learning).Otro trabajo, concerniente con ataques, según Delman (2004), fue sobre la transposición y permutación de cifrado, en algunos casos, aplicado a mono-alfabetos, entre ellos destacan: Clark (1994), Matthews (1993), Griindlingh & Van Vuuren (2002).

En este mismo contexto, Reddaiah (2019), realiza un estudio sobre GAs aplicado a la criptografía en el campo de e-commerce (comercio electrónico), donde se presentan funciones robustas en el procesamiento, usando diferentes tipos de GAs que son de gran importancia en el campo de la criptografía, ya que, proporcionan seguridad en los datos. En dicho trabajo,

se discuten las ventajas y desventajas de las funciones GAs presentadas, concluyendo que esta tendencia es de gran importancia para proporcionar seguridad, aunque el tamaño de la clave es reducido por GAs, produce buena seguridad para los datos. De otro modo, el manejo de claves se convierte en una difícil tarea. Una detallada descripción se obtiene sobre distintas funciones genéticas que son soportadas por la criptografía.

Además, Rodríguez (2020) propone un algoritmo criptográfico de tipo simétrico para texto, que aplica la filosofía de los Algoritmos Genéticos, la Entropía y la Aritmética Modular, empleando una metodología experimental dentro de un sistema determinista, el cual redistribuye y modifica los parámetros y fases del algoritmo genético que afectan directamente su comportamiento y se establece un cifrado independiente para la clave auxiliar haciendo uso de una clave principal, encargada de aumentar la seguridad; mostrando una comparación de resultados, con respecto a los algoritmos criptográficos: DES (Data Encryption Standard), RSA (Rivest, Shamir and Adleman) y AES (Advanced Encryption Standard), exponiendo factores como tiempo de ejecución, escalabilidad, tamaño de la clave, entre otros; llegando a demostrar que el algoritmo propuesto tiene un buen desempeño en estos términos (Rodríguez, 2020); otro trabajo similar es el presentado por: Nagaraj y Srinadth (2015), quienes proponen el uso de la encriptación de datos en caso de privacidad, para prevenir descubrimiento o confidencialidad durante proceso de las comunicaciones; presentando un nuevo método basado en el Teorema de Euler (Euler's Totient theorem) para producir un conjunto de números encriptados, que posteriormente generará la clave que se agregará a los datos encriptados, antes de la transmisión y desencripción, poder verificar la autenticación.

Por último, en la actualidad, los algoritmos de cifrado de datos se dividen en tres grupos (Progress Software Corporation, et. al. , 2020-2022): (1) Cifrado de texto o claves. Por ejemplo: DES, AES, RSA, MD5, SHA1, entre otros. (2) Cifrado de archivos (mediante protocolos). Por ejemplo: PGP, S/MIME, SSL, TSL, SSH, entre otros. (3). Encriptación para transferencia. Por ejemplo: FTPS, SFTP, SSL, AS2, AS3, AS4, HTTPS, MFT.

Es por ello, que en este trabajo de investigación, se presentan tres propuestas para el cifrado de datos, las cuales, son basadas en el Cifrado Caesar usando valores de la tabla ASCII. Además, se introduce el uso de métodos aleatorios con un algoritmo genético (para garantizar que el alfabeto Caesar esté bien revuelto); así como, un nuevo concepto, aquí denominado Pseudo-Hexadecimal, que lleva como propósito inyectar ruido al paquete de cifrado generado.

Cabe aclarar que el presente trabajo es un reporte técnico de la investigación, debido a que, al momento solamente se presentan resultados preliminares, puesto que, falta concluir algunos experimentos que permitirán demostrar o sugerir, que el cifrado/descifrado usando: Noised Random Pseudo-Hexadecimal, es seguro e indescifrable por las herramientas de software libre y/o comerciales que circulan en la actualidad; y de este modo, se justifica el presente proyecto, ya que, además de aportar evidencia empírica, se presentan nuevas propuesta para cifrado de datos.

## 2.- Objetivos

El objetivo principal de esta investigación, es llevar a cabo tareas de cifrado/descifrado de datos, utilizando el modelo estándar del Algoritmo de César (Caesar); comparándolo con tres nuevas propuestas basadas en Caesar, denominadas en este trabajo como: Random Caesar I, Random Caesar II y Noised Pseudo-Hexadecimal GAs; todo con el propósito de hacer la presentación de dichos algoritmos, y observar, si la introducción de ruido en el cifrado de datos, usando un nuevo concepto denominado: Pseudo-Hexadecimal, permite impedir que software especializado para descrifrado de datos traduzca la información cifrada, o en su defecto, pueda demorar o retardar un poco el descifrado de datos.

## 3.- Materiales y Métodos

En esta sección, se muestra una breve descripción de cada uno de los materiales, métodos, técnicas, algoritmos y métricas utilizados(as) en el desarrollo del presente trabajo. Cabe mencionar, que todo lo expuesto en este apartado, se encuentra descrito desde el punto de vista como fueron desarrollados; y todos los títulos de técnicas o métodos que llevan entre

paréntesis: NUEVA PROPUESTA; son consideradas(os) nuevas aportaciones, debido a que no se habían utilizado antes del desarrollo del presente proyecto, o al menos no con el mismo propósito que aquí se presenta.

### 3.1.- Materiales Utilizados

En esta sección se describen los materiales utilizados durante el proceso de experimentación de la presente investigación; como es el caso de: hardware, software y datos.

#### *3.1.1.- Hardware y Software*

En el presente trabajo de investigación, se llevaron a cabo dos fases en la etapa de implementación y experimentación. En la primera fase, se utilizó un equipo de cómputo marca Acer con 1 GB de memoria RAM y una velocidad de procesamiento a 1 GHz; con espacio disponible en disco duro de 300 GB. El sistema operativo utilizado fue Windows 7; y se instaló el lenguaje de programación: Python 3, para la implementación del software. En la segunda fase, se utilizó un dispositivo móvil, marca Hyundai con 1 GB de memoria RAM y velocidad de procesamiento a 1 GHz; con espacio disponible en almacenamiento interno de 14 GB. El sistema operativo utilizado fue Android 8.1; y se instaló la aplicación Pydroid3 para implementación con lenguaje Python 3. Al final, se hizo una prueba comparativa entre los resultados y/o código fuente generados, y no se observó diferencia significativa. Los algoritmos implementados funcionaron en las mismas condiciones y con comportamientos equivalentes.

#### *3.1.2.- Datos Utilizados*

Los datos utilizados en esta investigación, fueron mensajes de texto cifrados generados por cada uno de los algoritmos de cifrado implementados, de manera separada. Con dicha información, se diseño una muestra de entrenamiento, por experimento (es decir, para cada algoritmo de cifrado implementado); que incluye en cada fila o tupla, un patrón de entrenamiento, que consiste en el mensaje cifrado siendo etiquetado o clasificado con el texto del mensaje original. Las columnas o atributos de la muestra, es la extensión máxima del mensaje más grande, generalmente 256 posiciones (un caracter por posición), excepto para patrones de entrenamiento cuyo contenido se trataba de un archivo completo (en formato UTF-8), en este caso, tenía más posiciones; y también, para caso de los contenidos cifrados con código hexadecimal o Pseudo-Hexadecimal, cada columna o atributo, contiene dos caracteres por posición, para poder almacenar el código cifrado, por ejemplo: FF. Posteriormente, a cada muestra de entrenamiento, se le aplicó validación cruzada con cinco repeticiones (tomando 20% de cada clase como muestra de control y 80% como muestra de entrenamiento).

### 3.2.- Métodos Empleados

En esta sección se describen las técnicas y/o métodos; así como, algoritmos utilizados durante el proceso de experimentación en la presente investigación; como es el caso de: generalidades de las funciones de distancia o métricas; la regla del vecino más cercano (1-NN), algoritmo genético para alfabeto con cifrado ruidoso (Noised Cyphered GAs Dictionary) y los algoritmos de cifrado: SHA1, MD5, Cifrado del César (Caesar), Random Caesar I, Random Caesar II y Noised Pseudo-Hexadecimal GAs.

#### *3.2.1.- Cifrado con algoritmos basados en Tablas Hash*

Una función hash, según Lowery (2023), toma un valor de entrada (por ejemplo, una cadena) y regresa un valor de longitud fijada. Una función hash ideal tiene las siguientes propiedades: Es muy rápido; puede regresar un enorme rango de valores hash; genera un hash único para cada entrada única (sin colisiones); genera valores hash disimilares para valores de entrada

similares valores de hash generados no tienen un patrón discernible en su distribución. Ninguna función de hash ideal existe, por supuesto, pero cada uno apunta en operar tan cerca de lo ideal como sea posible. Sabiendo que (la mayoría) de las funciones hash regresan valores de longitudes fijadas y el rango de valores es, por lo tanto, restringido, esa restricción puede ser prácticamente ignorada. El número de valores posibles que pueden ser regresados por una función hash de 256-bit, por ejemplo, es aproximadamente el mismo que el número de átomos en el universo. Una función hash debería generar valores de hash distintos impredeciblemente para cualquier valor de entrada. Por ejemplo, toma las siguientes dos oraciones muy similares: “The quick brown fox." y "The quick brown fax.". Podemos comparar los valores hash de MD5 generados de cada una de las dos oraciones: “2e87284d245c2aae1c74fa4c50a74c77” y “c17b6e9b160cda0cf583e89ec7b7fc22”, respectivamente. Por lo tanto, se generaron dos hashes muy distintos para dos frases similares, lo que es una propiedad útil tanto para la validación como para la criptografía. Esto es un corolario de distribución: los valores hash de todas las entradas deberían ser esparcidas igualmente e impredeciblemente a lo largo de todo el rango de valores hash posibles. Las funciones hash comúnes, fueron diseñados por matemáticos y científicos de computación. En el transcurso de nueva investigación, algunos han mostrado tener debilidades, aunque todas son consideradas lo suficientemente buenas para aplicaciones nocriptográficas (Lowery, 2023).

#### *3.2.1.1.- MD5*

El MD5 o algoritmo de resumen de mensajes, es un protocolo criptográfico utilizado para autenticar mensajes, verificar contenido y firmas digitales; basado en una función hash para verificar si un archivo enviado coincide con el recibido en la transmisión de datos. En sus inicios, se utilizaba en cifrado de datos, y actualmente, puede encontrarse su uso para autenticación (Freda A., 2022). La función de hash MD5 produce un valor de hash de 128-bit. Fue diseñado para uso en criptografía, pero las vulnerabilidades fueron descubiertas en el transcurso del tiempo, así que ya no es recomendada para ese propósito. Sin embargo, todavía es usado para particionamiento de base de datos y sumas de control de computación para validar transferencias de archivos. Información detallada acerca de este algoritmo, lo podemos encontrar en (Willa, 2018 ; Lake, 2023).

#### *3.2.1.2.- SHA1*

El algoritmo SHA1 o Secure Hash Algorithm (Hash Seguro), parte de la familia SHA, de funciones que sirve para generar códigos hash. Desarrollado por el Instituto Nacional de Estándares y Tecnología con propósito de tener un estándar federal de procesamiento de información en EE. UU. (KeepCoding, 2023). La primera versión del algoritmo fue SHA-1, y luego seguido por SHA-2. Mientras que MD5 produce un hash de 128-bit, SHA1 genera un hash de 160-bit (20 bytes). En formato hexadecimal, es un entero de 40 dígitos de largo. Como MD5, fue diseñado para aplicaciones de criptología, pero pronto se le encontró vulnerabilidades también. Hoy en día, ya no es considerado ser menos resistente de atacar que MD5. Un uso típico de las funciones hash es hacer verificaciones de validaciones. Un uso frecuente es la validación de colecciones comprimidas de archivos, tales como archivos .zip o .tar; dado un archivo y su valor de hash esperado (comúnmente referido como una suma de control), puedes hacer tu propio cálculo de hash para validar que el archivo que recibiste está completo y no corrupto (Lowery & Pereyra, 2023). Con respecto a más detalles acerca del funcionamiento para el algoritmo SHA1, se puede encontrar en las fuentes (Lowery & Pereyra, 2023 ; PortalCripto, 2023).

### 3.2.2.- Cifrado por sustitución o desplazamiento

#### *3.2.2.1.- Cifrado del César (Algoritmo Caesar)*

El cifrado del cesar es uno de los más simples, usado por "El César" para comunicarse con sus generales, es un tipo de cifrado por sustitución en el que un símbolo del texto plano es sustituido por otro símbolo que se encuentra k posiciones adelante en el alfabeto. Por ejemplo, si suponemos que el alfabeto es el alfabeto del español y tenemos como texto plano “holaquetal”, y suponemos que k = 1 entonces cada letra se reemplazaría por la siguiente en el alfabeto. De esta forma la ‘h’ se convertiría en ‘i’, la ‘o’ en ‘p’ y asi sucesivamente hasta obtener el texto encriptado “ipmbrvfubm” (Gómez S. , et.al. 2012). Formalmente se puede definir el Cifrado del César como se muestra enseguida: C[i] = S[i] + K (mod N). Donde: S[i] corresponde al carácter en la posición i del texto plano; C[i] corresponde al carácter i del texto encriptado y N corresponde a la cantidad de símbolos del alfabeto. La desencripción sería de una forma similar: C[i] = S[i] - K (mod N). El problema evidente que tiene este método es que un mismo símbolo en el texto plano, se encripta al mismo símbolo en el texto encriptado. De esta forma si un criptoanalista sabe que se utilizó este método podría simplemente ver cuál es la letra más repetida en el texto encriptado y sabiendo que en el alfabeto español la letra "e" es la que más se repite podría simplemente hacer la resta entre estas dos letras y hallar K fácilmente (Luciano & Prichett, 1987 ; Gómez et.al, 2012).

Según Barranco y Galindo (2022), el algoritmo de César, es un criptosistema o cifrado por desplazamiento. Entonces: Sea P=C=K=Z26. Para cada K ∈K, definimos: eK(x)=x+K mód 26 ; dK(y)=y−K mód 26. Como curiosidad, se sabe que para la clave particular K=3, el criptosistema se llama a menudo el cifrado César debido a su uso reportado por Julio César (Barranco & Galindo, 2022). Este criptosistema es realmente sencillo: a cada letra del alfabeto le corresponde un número módulo 26 , al que aladimos la clave módulo 26 y obtenemos la letra asociada. Ciframos todo el mensaje aplicando esto a cada una de las letras. Para descifrar, hacemos lo mismo, pero tomamos la diferencia en lugar de sumar (Barranco & Galindo, 2022). Ejemplo: Tomamos la clave K=11, y el texto plano: wewillmeetatmidnight. Suponiendo que: [A=0, B=1, C=2, D=3, E=4, F=5, G=6, H=7, I=8, J=9, K=10, L=11, M=12, N=13, O=14, P=15, Q=16, R=17, S=18, T=19, U=20, V=21, W=22, X=23, Y=24, Z=25]. Primero, convertimos el texto plano en sus números asociados en Z26, es decir, de 0 a 25 (Barranco & Galindo, 2022): [22,4,22,8,11,11,12,4,4,19,0,19,12,8,3,13,8,6,7,19]. Ahora añadimos la clave (sumamos el valor K=11) a cada valor en Z26, tenemos:[7,15,7,19,22,22,23,15,15,4,11,4,23,19,14,24,19,17,18,4]. Por último, convertimos estos números en letras y obtenemos el cibertexto: HPHTWWXPPELEXTOYTRSE (Si el número es mayor que mod N , se le resta N. En el ejemplo: mod 26: 33 - 26 = 7 y 30 - 26 = 4). Sin embargo, este cripto sistema no es seguro ya que puede ser criptoanalizado fácilmente mediante la búsqueda exhaustiva de claves, es decir, buscando con todas las claves posibles. Como sólo hay 26 claves, podemos probar todas las reglas de descifrado posibles hasta obtener un texto plano significativo. Además, se puede demostrar que sólo se necesita una media de 13 intentos hasta que encontramos la clave y la regla de descifrado adecuadas (Stinson D.R., 2005 ; Barranco & Galindo, 2022).

#### *3.2.2.2.- Random Caesar I: Cifrado del César Aleatorio #1 (NUEVA PROPUESTA)*

Esta nueva propuesta, es una variante del Algoritmo de César, descrito previamente, pero difieren en la extensión del alfabeto (mod 26) y que dicho alfabeto se selecciona aleatoriamente (sin reemplazo). Existen dos versiones de este algoritmo. La primer versión (Random Caesar I) usa los 255 caracteres del alfabeto ASCII (Código Estadounidense Estándar para el Intercambio de Información) y la versión II, solamente utiliza el rango de: 32 a 126 del ASCII. El algoritmo consiste en dos partes: La primera parte, es el cifrado (parcial) del texto a ocultar, donde se utiliza un K (o SHIFT) diferente para cada caracter que se desea cifrar (denominado Ki). La segunda parte, corresponde al empaquetado del mensaje, que incluye el mensaje cifrado dos veces (para efecto de inyectar "ruido"), una en forma de caracter, y otra en tipo de datos ordinal (entero ASCII). También, se agrega al mensaje cifrado, el vector de Ki, para poder descifrar.

Formalmente, se puede definir como se muestra enseguida. Para el cifrado parcial utilice: C[i] = S[i] + K[i] (mod N) ; y para descifrado parcial: D[i] = C[i] - K[i] (mod N). Donde: S[i] corresponde al carácter en la posición i del texto plano. El vector C[i] corresponde al carácter i del texto encriptado. La variable D[i] corresponde al carácter i del texto desencriptado. El vector entero Ki son los desplazamientos a realizar por cada caracter; y la variable N corresponde a la cantidad de símbolos del alfabeto (en caso de esta primer variante, se utiliza un rango de 0 hasta 255, que corresponden al ASCII). Por ejemplo: Suponiendo que para la primera parte (del cifrado parcial) tenemos los vectores: C[i] = [Q, A, N, E, U, 2, *, B, O, X, ~, □ cuyo ordinal: Ord[i]= [81, 65, 78, 69, 85, 50, 42, 66, 79, 88, 126, 255] ; y su desplazamiento: K[i] = [12, 3, 1, 3, 2, 40, 7, 2, 1, 12, 0]. Dichos vectores: C[i] y K[i] han sido seleccionados de manera aleatoria; C[i] aleatoria sin reemplazo y K[i] aleatoriamente con reemplazo. Posteriormente, suponiendo que no deseamos incorporar claves públicas o privadas, sino que, solamente se desea encriptar un texto, el cual es tomado como clave privada desde una entrada de datos, siendo este texto a cifrar el siguiente: S[i] = [B, U, E, N, O] ; cuyo vector de ordinales es: Ord(S[i]) = [66, 85, 69, 78, 79]. Entonces, extraemos del vector de desplazamiento Ki original, los valores que corresponden a cada caracter integrado en S[i] teniendo un nuevo subvector para K'[i] = [ 2, 2, 3, 1, 1]. Al sumar los ordinales de S[i] con los valores del nuevo subvector K'[i], tenemos que: Ord'(C[i]) = ( Ord(S[i]) + K'[i] ) ; tenemos como resultado: [68, 87, 72, 79, 80] y convirtiéndose a valores de la tabla ASCII, tenemos el texto cifrado (parcialmente), en una primera fase: Cifrado Parcial = C'[i] = [D, W, H, O, P]. Posteriormente, se aplica una segunda fase o parte del algoritmo, que consiste en hacer un empaquetado, que lleva como contenido, los vectores intercalados, definido formalmente como: MensajeCifrado = C'[i]+K'[i]+Ord'(C'[i]) ; donde el operador + refiere a la función concatenar; aunque en este último vector: Ord'(C'[i]), puede incluir o no la suma de su desplazamiento u otro ordinal que corresponda a un caracter (por ejemplo, seleccionado aleatoriamente, con o sin reemplazo), según convenio previo con el criptoanálista (ya que, el propósito de este vector es inyectar ruido); dando lugar al siguiente mensaje cifrado final: [ D, 2, 68, W, 2, 87, H, 3, 72, O, 1, 79, P, 1, 80 ]. Que en realidad, al ser almacenado, previo convenio con el criptoanálista, se mostraría en un primer formato como: D2DW2WH3HO1OP1P; o en un segundo formato podríamos apreciar el ordinal (que incluye desplazamiento u otro caracter) en una o dos cifras: D2**68**W2**87**H3**72**O1**79**P1**80**. Por último, para descrifrar, solamente se separan las partes del empaquetado y se resta K'[i] a C'[i].

### *3.2.2.3.- Random Caesar II : Cifrado del César Aleatorio #2 (Random Caesar I Mejorado, NUEVA PROPUESTA).*

Los algoritmos: Random Caesar I y Random Caesar II, trabajan utilizando la misma lógica computacional, pero difieren en la extensión del alfabeto de cifrado; ya que, la primer versión (Random Caesar I) usa los 255 caracteres del alfabeto ASCII (Código Estadounidense Estándar para el Intercambio de Información) y la versión II, se inició utilizando solamente caracteres, dentro del rango de: 32 a 126 del ASCII. El propósito de esta modificación, fue debido a que, la primer versión, permitía el proceso de cifrado correctamente, pero al momento de descifrar los datos, en algunos casos se detectó error, se cree, ocasionado por pérdida de información en caracteres del ASCII que no son imprimibles y/o corresponden a teclas de función, los cuales, no fueron capaces de mostrarse en la pantalla o campos de texto de la interfaz del programa de cómputo utilizado, y por ende, algunos caracteres no fueron guardados en archivos de las pruebas realizadas, lo cual, producía el error. Por lo tanto, acotando el rango del código ASCII, sin incluir los caracteres no imprimibles y/o de teclas de función, resultaba muy prometedor al inicio. Sin embargo, no se observaba ruido en la salida parcial del mensaje cifrado, es por ello, que se incluyeron otros caracteres para inyectar ruido, quedando un límite de 120, como posibles caracteres candidatos, dentro del rango comprendido entre los valores: desde el 30 al 150 del ASCII.

Este algoritmo: Random Caesar II, también consiste en dos partes: o fases, cuya implementación es la misma que su sucesor (Random Caesar I), excepto en la segunda parte, donde debe realizar convenio previo con el criptoanálista, esta decisión se omite en la versión II. En su lugar, se decide duplicar el mensaje de cifrado parcial dentro del empaquetado (en ordinal o caracter, sin desplazamientos) y se acompaña del vector Ki, que incluye los desplazamientos generados de manera aleatoria con reemplazo. Cabe aclarar, que el modo de selección del alfabeto de cifrado (C[i]) no cambia, es decir, se sigue realizando de manera aleatoria sin reemplazo. El algoritmo: Random Caesar II, se puede definir, de manera similar a su sucesor, tal como

se muestra enseguida. Parte 1 del algoritmo (Cifrado Parcial): C[i] = S[i] + K[i] (mod N) ; Descifrado Parcial: D[i] = C[i] - K[i] (mod N). Donde: S[i] corresponde al carácter en la posición i del texto plano. El vector C[i] corresponde al carácter i del texto encriptado. El vector D[i] corresponde al carácter i del texto desencriptado. El vector Ki son los desplazamientos a realizar por cada caracter. La variable N corresponde a la cantidad de símbolos del alfabeto (inicialmente: de 32 a 126 ; y en la versión mejorada: de 130 a 150). Del mismo modo, en la segunda parte o fase, se utiliza para empaquetado del mensaje cifrado: EmpaquetadoFinal = C'[i]+K'[i]+Ord'(C'[i]) ; donde el operador + refiere a la función concatenar; el vector C'[i] es el mensaje cifrado; mientras que en K'[i] se encuentran los valores del desplazamiento y el vector Ord'(C'[i]) son los valores ordinales, byte o su transformación en tipo "char".

### 3.2.3.- Cifrado con Inteligencia Artificial

En esta sección se describen algunos métodos, técnicas, métricas y algoritmos, que son utilizados en las áreas de Inteligencia Artificial y Reconocimiento de Patrones; los cuales, están relacionados con el presente trabajo de investigación.

#### 3.2.3.1.- Información Básica

Debido a que, en este trabajo de investigación, se propone el uso de Inteligencia Artificial y Reconocimiento de Patrones, recomendando el uso de la Regla 1-NN para clasificación de patrones (que en este caso particular, los patrones son mensajes cifrados convertidos a tipo de dato ordinal entero), es por ello, que se considera necesario realizar la descripción del uso de distancia empleada en 1-NN, descripción del algoritmo de la regla del vecino más cercano; así como, el algoritmo genético utilizado para generación del alfabeto a utilizar en los algoritmos de cifrado de datos; tal como se describe en las siguientes secciones.

##### *3.2.3.1.1.- Generalidades Respecto A Las Funciones De Distancia*

Según Rangel (2002), una función de distancia debe cumplir con tres propiedades: (1) 'No negatividad', es decir, la distancia entre Xi & Yi, debe ser mayor o igual que cero [ d(x,y) >= 0 ]. (2) 'Simetría', significa que la distancia entre Xi & Yi, debe ser igual que la distancia entre Yi & Xi [ d(x,y) = d(y,x) ]. (3) 'Desigualdad Triangular', ello quiere decir que la distancia entre Xi & Yi debe ser menor o igual que la suma de las distancias entre Xi & Zi , comprendiendo a Zi & Yi , de modo tal que satisfaga: d(x,y) <= d(x,z) + d(z,y).

##### *3.2.3.1.2.- Distancia Euclidiana*

La Distancia Euclidiana es una de las funciones o métricas más utilizadas en la regla del vecino más cercano (Hart P.E. , 1968), y su ecuación es la siguiente:

$$D_e(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \qquad (3.2.3.1.2.1)$$

Donde: 'DE' representa el resultado de la 'Distancia Eucledíana'; 'xi' & 'yi' son los componentes i-ésimos del vector 'x' & 'y' , respectivamente; y 'd' es el tamaño de los vectores 'x' & 'y' (Rangel E. , 2002; Rangel E. , 2022).

***3.2.3.1.3.- La Regla Del Vecino Más Cercano (Regla 1-NN [Nearest Neighbor] )***

La Regla 1-NN (Cover & Hart ,1967) es utilizada generalmente en aspectos de clasificación. También se ha utilizado en algunas técnicas ó métodos, como en el caso del Método L o Leave-One-Out (Barandela R. & Gasca E., 2000). La referida Regla NN (1-NN), ya ha sido definida y evaluada por otros autores (Cover & Hart ,1967; Fix & Hodges, 1952; Gongde et. al. , 2004; Hand et. al. , 2001; Wang H., 2002). Sin embargo, a continuación se describe el algoritmo, de la misma manera que ha sido empleado en otras investigaciones (Rangel E., 2002; Barandela et. al. , 2003, Rangel R. , 2018; Rangel E. , 2019; Rangel E. , 2022): (a) Primero, se almacena la muestra de entrenamiento en memoria. (b) Cuando llega un patrón nuevo 'X', sin etiqueta o desconocido, se busca en la muestra de entrenamiento su vecino más cercano; es decir, el patrón de entrenamiento que produzca la menor distancia con 'X'. Para ello, se tiene que calcular la 'Distancia Euclidiana' entre 'X' y cada uno de los patrones de entrenamiento (ver Ecuación 3.2.3.1.2.1), para observar cual distancia obtenida es más pequeña. (c) Por último, se asigna a 'X' la etiqueta de su vecino más cercano, y se clasifica como miembro de la clase de su vecino más cercano (Rangel E. , 2002 ; Cover T. M. & Hart P. E. , 1967; Rangel E. , 2022).

***3.2.3.1.4.- Noised Cyphered GAs Dictionary: Algoritmo Genético Para Generar Alfabeto/Diccionario Con Cifrado Ruidoso (NUEVA PROPUESTA)***

El proceso de este método, por tratarse de un algoritmo genético, cuenta con cuatro fases: (i) Selección ; (ii) Cruce ; (iii) Mutación ; y, (iv) Evaluación. El algoritmo: Noised Cyphered GAs Dictionary, comienza seleccionando (de manera aleatoria con remplazo) un vector denominado: ABC, que será considerado el alfabeto inicial (cuyos valores ordinales ASCII, se recomiendan dentro del rango: 30 a 150). Posteriormente, se genera de manera aleatoria (con reemplazo) una muestra de entrenamiento (ME), de dos clases, con tamaño 'n' filas (que representarán los cromosomas) y (m+1) columnas (que corresponde a los atributos de cada cromosoma, que comúnmente son conocidos como bits). Donde 'n' debe ser un valor par, ya sea, definido por el usuario o de manera aleatoria (se recomienda usar n = 100 ejemplares), espacio en el cual, cada renglón será ocupado por un cromosoma, que consiste en un patrón aleatorio que refiere a una secuencia de ordinales ASCII que se entiende es una propuesta (de solución) para sustituir el alfabeto que se usará para cifrado o descifrado de datos. El valor 'm' se refiere al número de caracteres ordinales ASCII a considerar (por ejemplo: m=255 si se usa la tablatura completa; m=120 si se usa un rango de 30 a 150, para incorporar ruido; o utilice m=94 posiciones si se desea utilizar valores sin ruido dentro del rango: 32 a 126). El parámetro +1, del tamaño (m+1), se refiere a que en esta última columna, se almacenará el valor de la etiqueta de clase (por ejemplo: 0 para el valor inicial, sin etiqueta, al ser creada la ME por primera vez. El valor de 1, indica clase #1, refiriendo a los patrones más cercanos al vector del alfabeto ABC; y, una etiqueta con valor de 2, refiere a clase #2, es decir, los patrones menos cercanos o parecidos al vector ABC). Entonces, empieza el proceso de selección.

***3.2.3.1.4.1.- Proceso De Selección***

El proceso de selección del algoritmo genético: Noised Cyphered GAs Dictionary, inicia cuando ya se tiene generado un vector inicial: ABC; y, existe una ME inicial, que tiene aún cromosomas con algunas etiquetas de valor=0. Posteriormente, se selecciona (de manera secuencial) un par de cromosomas (filas o patrones) y se calcula la Regla 1-NN (con distancia euclideana), del vector ABC con estos dos patrones de entrenamiento. El cromosoma que corresponda al vecino más cercano de ABC, se le actualiza el valor de la etiqueta con 1, mientras que se asignará valor de 2, a la etiqueta del cromosoma que no fue más cercano al vector ABC. Este proceso se repite hasta agotar los cromosomas de la ME, con el propósito de que todos los cromosomas tengan identificación de clase, no importa en generación se lleve a cabo ello; por ejemplo, en una segunda generación, los valores siempre deben ser actualizados, aún cuando no tengan etiquetas con valor=0). Ahora, deberá crear un vector vacío denominado xABC, de la misma longitud que ABC (en la segunda generación del algoritmo, en adelante, ya no será necesario crearlo, solamente será actualizado con los nuevos valores). El vector xABC, representará al alfabeto para

cifrado de datos, mientras que ABC, es el alfabeto inicial, que podrá usarse para el descifrado de datos. Entonces, utilizando la ME, deberá seleccionar de manera aleatoria un cromosoma de clase #1, para asignar al vector ABC, ocupando su lugar, al ser sustituidos o actualizados sus valores. También, deberá sustituir o actualizar los valores del vector xABC, asignando los valores de un cromosoma de clase #2, que haya sido seleccionado previamente, de manera aleatoria. Para finalizar la fase de selección, de la actual generación, deberá eliminar de la ME, en forma aleatoria, el 50% de los cromosomas de la clase #1; y también, de manera aleatoria, deberá eliminar de la ME, el 50% de los cromosomas de la clase #2.

#### *3.2.3.1.4.2.- Proceso De Cruce*

El proceso de cruce del algoritmo genético: Noised Cyphered GAs Dictionary, inicia una vez que se han actualizado los valores de los vectores: ABC y xABC; así como, han quedado en la ME espacios vacíos para poder almacenar el 50% de nuevos cromosomas (que serán hijos) de cada clase. El cruce, se inicia con un proceso secuencial (y paralelo), seleccionando un par de cromosomas de diferente clase. Para cada par de cromosomas, se decide aleatoria, el método a utilizar para su cruce. El algoritmo: Noised Cyphered GAs Dictionary, solamente permite el cruce por la mitad y/o en dos puntos (extremos). Por ejemplo, suponiendo que se ha decidido, para el primer par de cromosomas (uno de clase #1 y otro de clase #2), emplear cruce por la mitad, teniendo como patrones o cromosomas: ME0Clase1 = [65, 66, 78, 32, 68, 33] y ME50Clase2 = [91, 70, 67, 99, 40, 41] ; el resultado del cruce será: ME26Clase1 = [65, 66, 78, 99, 40, 41] y ME51Clase2 = [91, 70, 67, 32, 68, 33]; a este par de hijos cromosomas se asignará etiqueta con valor de 0. En cambio, si se hubiera decidido utilizar el método de cruce por los extremos (en dos puntos); el resultado sería: ME26Clase1 = [40, 41, 78, 32, 91, 70] y ME51Clase2 = [68, 33, 67, 99, 65, 66]; siendo ambos vectores etiquetados como clase 0. Este proceso de crucé se repite, hasta agotar cada par de cromosomas de diferente clase en la ME, teniendo el 50% de cromosomas cuya etiqueta sea valor=0.

#### *3.2.3.1.4.3.- Proceso De Mutación*

El proceso de mutación del algoritmo genético: Noised Cyphered GAs Dictionary, se realiza sobre los vectores ABC y xABC; y también, a los cromosomas de la ME. El método de mutación, se decide aleatoriamente, tanto para los vectores: ABC y xABC; como para cada cromosoma de la ME. El algoritmo: Noised Cyphered GAs Dictionary, solamente permite mutación por intercambio y/o por sustitución; y solo se aplica a un número determinado de bits (columnas: 'm') del cromosoma. Por ejemplo, en supuesto de que el vector o cromosoma de la ME sea el siguiente: VMEi = [65, 66, 78, 91, 70, 67] y suponiendo que se desea aplicar mutación por intercambio a un solo solo bit (posición o columna del cromosoma), ello afectará a dos localidades. Ello quiere decir, que deberá seleccionar de manera aleatoria, el par de localidades de bits a intercambiar. Suponiendo que en el sorteo se seleccionan las localidades o bits: 1 y 4 (aclarando que el conteo de posiciones inicia desde 0), entonces el vector o cromosoma mutado por intercambio será el siguiente: Muted-VMEi = [65, 70, 78, 91, 66, 67], se habrá intercambiado el ordinal 66 por 70 en el patrón del cromosoma. En cambio, si el método de mutación hubiera sido por sustitución, en este caso, se elige el número de bits a mutar. Si se selecciona un solo bit, ello afectará solamente a una localidad del cromosoma. Suponiendo que de manera aleatoria, se ha seleccionado la localidad 1, se tendrá que mutar el segundo bit del cromosoma, sustituyendo dicho valor con otro número ordinal generado de manera aleatoria (solo deben considerarse valores dentro del mismo rango con que ha sido creada la ME), quedando como resultado: Muted-VMEi = [65, 66, 79, 91, 70, 67], entendiendo que el valor 78 de la segunda localidad del cromosoma fue sustituido por uno nuevo, y durante el sorteo aleatorio fue seleccionado el ordinal 79. En la literatura, la mutación es posible percibirla en un contexto de bits con código binario, intercambiando un valor de 0 por 1 (o viceversa). Sin embargo, en el contexto de los números ordinales, no fue posible realizar ello en el algoritmo: Noised Cyphered GAs Dictionary, por tal razón, se sustituye por otro valor ordinal, generado de forma aleatoria. Otra consideración, es que la mutación a la ME, no se recomienda realizar a todos los cromosomas o patrones de entrenamiento, sino solamente aplicar, a un porcentaje específico, por ejemplo: 10% o 20% de cada grupo de clases, instanciados en la ME.

*3.2.3.1.4.4.- Proceso De Evaluación*

Después de haber realizado de manera iterativa 100 generaciones, se procede a evaluar con la función de aptitud: P = ( ( SUM ( DIF ( ABCi , xABCi ) ) * 100 ) / m ). Donde: El parámetro 'm' refiere al número de bits o posiciones (columnas) del vector: ABC o xABC (que deben ser del mismo tamaño). La función DIF, es una métrica de similaridad/disimilaridad que permite a evaluar cuanto diferentes son cada uno de los bits (o posiciones) de los vectores: ABC y xABC; esta función regresa valor de 1 si el valor de la localidad i del vector ABC es igual que xABCi ; si son diferentes regresa valor de 0. La función SUM permite sumar todos los resultados de cada evaluación realizada por la función DIF. Si el resultado de SUM es igual al número de columnas (bits) de ABC o longitud de xABC, se entiende que los vectores no son aptos todavía, porque podrían ser iguales. El resultado de la función de aptitud P, indica el porcentaje de error, es decir, el porciento en que los vectores: ABC y xABC tienen de similitud. El valor ideal para terminar con el procedimiento del algoritmo genético, es un valor de 0. En otro caso, se repite nuevamente el proceso de selección, cruce y mutación; otras 100 generaciones, partiendo con los valores actuales de ABC y xABC; así como, usando la misma ME generada por la última o previa generación de la aplicación del algoritmo genético. Al finalizar, se evalúa nuevamente la función de aptitud P, si regresa valor mayor que 0, se vuelve a repetir el proceso otras 100 generaciones, y así sucesivamente, hasta que el valor de P sea 0. Por último, se regresa como resultado los vectores: ABC y xABC, que refieren al alfabeto original y su alfabeto de cifrado, respectivamente; aunque ABC puede ser utilizado posteriormente como alfabeto para realizar el descifrado de datos. Finalmente, cabe aclarar, que no es obligatorio realizar 100 generaciones en cada proceso de evaluación, sino que, el número de generaciones puede ser definido de manera anticipada.

**3.2.3.2.- Noised Pseudo-Hexadecimal GAs: Algoritmo Genético Con Ruido Pseudo-Hexadecimal (NUEVA PROPUESTA)**

En términos generales, podría decirse que el algoritmo: Random Caesar II, y su sucesor: Noised Pseudo-Hexadecimal GAs, utilizan la misma lógica computacional; excepto en la selección del alfabeto inicial y/o alfabeto para cifrado de datos; ya que, utiliza un algoritmo genético para tales propósitos, y además, este último algoritmo, en lugar de trabajar con valores ordinales, sugiere el uso de valores hexadecimales, así como, un nuevo concepto denominado: Pseudo-Hexadecimal, para el cifrado de datos.

Recordando un poco, sobre el algoritmo: Noised Cyphered GAs Dictionary, donde se propone el cifrado/descifrado de datos, haciendo uso del método de sustitución. Es decir, para cifrar se busca una ocurrencia en el vector ABC y se sustituye por el valor correspondiente localizado en vector xABC. El proceso comienza, definiendo un texto de entrada a cifrar, y se busca cada caracter (ordinal ASCII) en cada posición del vector: ABC (el último generado por: Noised Cyphered GAs Dictionary), regresando el valor correspondiente a la misma posición o localidad del vector: xABC. Al final, se tiene como texto cifrado, la concatenación de caracteres encontrados en xABC. En caso del algoritmo: Noised Pseudo-Hexadecimal GAs, consiste en tres fases o etapas; y no es obligatorio utilizar ambos vectores: ABC y xABC para el cifrado de datos. Solo puede seleccionar uno de ellos como alfabeto inicial, para comenzar el proceso de desplazamiento siendo controlado por un vector denominado: Ki, lo cual, permite la generación de un nuevo vector, que contiene el alfabeto para cifrado de datos. Sin embargo, si queremos evitar el cálculo de los desplazamientos Ki, podemos hacer uso de los vectores: ABC y xABC, pero a este último vector, tendremos que aplicarle el concepto: Pseudo-Hexadecimal, para generar un cifrado parcial en dicho formato, y poder saltar a la fase o parte 3, del algoritmo: Noised Pseudo-Hexadecimal GAs; es por ello, que este algoritmo introduce este nuevo concepto (denominado: Psedo-Hexadecimal), que consiste en introducir ruido a algunos números hexadecimales, generando secuencias con otros caracteres del ASCII, parecidos a hexadecimal, pero no se encuentran en un rango válido (Por ejemplo: el hexadecimal del caracter N es 4e y se escribe como 0x4E ; y para inyectar ruido el valor Psedo-Hexadecimal sería: 0x0N, lo cual, en números hexadecimales no es válido), todo ello con la finalidad de conseguir que el texto cifrado sea descifrado con mayor dificultad.

Por lo tanto, Noised Pseudo-Hexadecimal GAs, es una variante del Random Caesar II, a pesar que coinciden en el alfabeto, porque utiliza solamente 120 caracteres del ASCII (mod 120), dentro del rango: 30 a 150; pero difieren porque en: Noised Pseudo-Hexadecimal GAs, se usan valores tipo hexadecimal; y se agrega ruido en formato: Pseudo-Hexadecimal, simulando los caracteres del ASCII como número hexadecimal. Otra diferencia con Random Caesar II, es la modificación del parámetro K[i]. Es decir, en lugar de sumar K[i], ahora se desplaza y genera un nuevo alfabeto de cifrado. Las partes o fases del algoritmo Noised Pseudo-Hexadecimal GAs, se describen enseguida:

PARTE 1: La primera parte o fase, consiste en utilizar el algoritmo genético: Noised Cyphered GAs Dictionary, para generar el alfabeto inicial y poder garantizar que exista una buen grado de aleatoriedad sin reemplazo o coincidencias.

PARTE 2: La segunda parte o fase, corresponde al cifrado (parcial) del texto a ocultar, donde se utiliza un segundo alfabeto (de cifrado), generado por desplazamiento K[i] (vector seleccionado aleatoriamente con reemplazo), el cual, debe ser, un valor diferente para cada caracter que se desea cifrar; o en su defecto, se puede seleccionar el vector xABC generado por: Noised Cyphered GAs Dictionary; y posteriormente, aplicar el concepto: Pseudo-Hexadecimal. Por ejemplo: Suponiendo que el alfabeto inicial (ya sea: ABC o xABC), convertido a caracter ASCII, es el siguiente vector: C[i] = [ Q , A , N , E , U , @ , ^ , B , O , X ] y su correspondiente hexadecimal es: Hex( C[i] ) = [ 51 , 41 , 4e , 45 , 55 , 40 , 5e , 42 , 4f , 58 ] . Del mismo modo, supongamos que el vector de desplazamiento, seleccionado aleatoriamente con reemplazo, es el siguiente: K[i] = [ 7 , 3 , 1 , 3 , 5 , 0 , 6 , 3 , 1 , 0 ] ; entonces al aplicar desplazamiento Ki sobre Ci, teniendo en cuenta que las posiciones del vector empiezan en 1 (es decir: C[1] = Q , C[2] = A , C[3] = N , C[4] = E , C[5] = U , C[6] = @ , C[7] = ^ , C[8] = B , C[9] = O y C[10] = X), obtenemos el vector de sustitución para cifrado: C' [i] = [ B , U , E , ^ , X , @ , N , Q , A , O ] ; lo anterior, porque el caracter Q se encuentra en la posición 1 de Ci (es decir: C[1]) y al sumar su desplazamiento Ki con valor de 7, entonces: 1+7=8, y el caracter ubicado en la posición C[8] es B; en cambio, el caracter A se encuentra en la posición 2 de Ci (es decir: C[2]) y al sumar su desplazamiento Ki con valor de 3, entonces: 2+3=5, y el caracter ubicado en la posición C[5] es U; y así sucesivamente. Este proceso de desplazamiento Ki, podría omitirse, si hacemos uso del vector: xABC, generado por: Noised Cyphered GAs Dictionary. Independiente de ello, al final, deberá convertirse el vector C'[i] (o xABC, según sea el caso), utilizando el nuevo concepto: Pseudo-Hexadecimal; lo cual, quedaría de la siguiente manera: C' [i] = [ 0x0B , 0x0U , 0x0E , 0x0^ , 0x0X , 0x0@ , 0x0N , 0x0Q , 0x0A , 0x0O ] y su equivalente en hexadecimal: Hex (C' [i] ) = [ 42 , 55 , 45 , 5e , 58 , 40 , 4e , 51 , 41 , 4f ]. Por lo tanto, supongamos que el texto de entrada a cifrar es: S[i] = [ B , U , E , N , O ] y su correspondiente hexadecimal es: Hex(S[i]) = [ 42 , 55 , 45 , 4e , 4f ] ; al ser buscado cada caracter i de S[i] dentro de C[i], se regresará su correspondiente en C'[i] o xABCi (según corresponda, en caso de haber omitido cálculo de desplazamiento Ki, se usará xABC); y se obtiene el resultado del cifrado parcial en formato Pseudo-Hexadecimal: CParcial[i] = ( [ 0x0Q , 0x0X , 0x0^ , 0x0E , 0x0A ] + RELLENO ). El vector de RELLENO, consiste en valores aleatorios hexadecimales y/o Pseudo-Hexadecimales, que tendrán que agregarse, en caso de que el vector S[i] sea menor que C[i] (o viceversa); ya que, en el algoritmo: Noised Pseudo-Hexadecimal GAs, se propone que los tres vectores a empaquetar en la tercera fase o etapa, sean del mismo tamaño, para evitar Identificar con facilidad, el orígen del vector que contiene el mensaje con cifrado parcial. Finalmente, deben extraerse solamente el par de caracteres hexadecimales (o Pseudo-Hexadecimales), regresando una salida similar a la siguiente: CifradoParcial = [ 0Q , 0X , 0^ , 0E , 0A ] + RELLENO.

PARTE 3: La tercera parte o fase, corresponde al empaquetado del mensaje final, que incluye los 120 caracteres del alfabeto en formato hexadecimal; adicionando otros 120 caracteres en formato: Pseudo-Hexadecimales (que corresponde al alfabeto de cifrado); y por último, se agrega el texto del cifrado parcial, también de 120 caracteres; pero si el mensaje es más corto, se rellena con hexadecimales y/o Pseudo-Hexadecimales, según se prefiera, pero deben ser seleccionados de manera aleatoria con reemplazo, para inyectar más ruido al mensaje. En caso de que el texto del mensaje a cifrar sea mayor de 120 caracteres, entonces se tendrán que rellenar los vectores: C[i] y C'[i] (o xABC si fuera el caso) con valores hexadecimales y/o Pseudo-Hexadecimales, usando selección aleatoria con reemplazo. El empaquetado debe realizarse intercalado. Por ejemplo: MensajeFinal = C[i] + C'[i]+ CifradoParcial[i] ; donde el operador + , significa concatenación.

### *3.2.4.- Método de estimación de error: Validación Cruzada o Método π (PI)*

Este método consiste en la aplicación de los siguientes pasos (Rangel E. , 2002, Rangel E, 2022): (1) Se extrae un grupo de patrones de entrenamiento 'ME Pi' de tamaño 'Pi'. (2) El modelo se entrena con la muestra de entrenamiento ('ME') sin incluir a 'ME Pi'. (3) El modelo se evalúa con 'ME Pi'. (4) El proceso se repite para: i = 1, 2, ..., ( n / p ). Donde: 'p' es el porcentaje utilizado para 'ME Pi' que en ocasiones se le conoce como muestra de control ('MC'). En esta investigación, la ME consiste en un conjunto de mensajes cifrados, teniendo como etiqueta de clase su cadena de texto del mensaje original. El modelo que evalúa la ME, es el módulo de descifrado de datos de cada algoritmo de cifrado señalado en la sección de métodos de la presente investigación.

## 4.- Experimentación, Resultados (Preliminares) y Discusión

En esta sección, se describe el desarrollo de la investigación realizada en el presente trabajo. Las muestras utilizadas en la experimentación, la mayoría de ellas fueron diseñadas manualmente y contienen datos tipo alfanuméricos (y en algunos casos, incluyen símbolos). Los patrones de entrenamiento, incluyen mensajes cifrados con Caesar, Random Caesar I, Random Caesar II y Noised Pseudo-Hexadecimal GAs. También, es necesario mencionar, que la experimentación de todas las técnicas y/o métodos utilizados en el presente trabajo, se llevó a cabo, evaluando con el método de estimación de error: Método π (PI) ó 'Validación Cruzada', siendo empleado con cinco repeticiones para cada muestra de entrenamiento [es decir, 80% para Muestra de Entrenamiento (M.E.); y 20% para Muestra de Control (M.C.)]. Los tipos de datos de las muestras de entrenamiento, utilizadas durante la experimentación, son cadenas de texto (con extensión máxima de 255 caracteres) que incluyen mensajes cifrados con Caesar, Random Caesar I, Random Caesar II y Noised Pseudo-Hexadecimal GAs. Las muestras de entrenamiento utilizadas, cada una corresponde a un algoritmo de cifrado distinto y se evaluaron de manera separada. Dichas muestras de entrenamiento, están clasificadas de acuerdo a su etiqueta, la cual corresponde al texto original; y el patrón de entrenamiento, es el mensaje cifrado que tiene como número de columnas o atributos, el tamaño del texto, un caracter por posición; excepto los mensajes cifrados con formato Pseudo-Hexadecimal, que tienen dos caracteres por cada posición de columna, para poder almacenar, por ejemplo: FF. Por lo tanto, se diseñaron los patrones de entrenamiento como cadenas de mensajes de texto (similares a contraseñas); y en algunos casos, se hizo pruebas, utilizando diversos archivos en formato texto (UTF-8), los cuales, estaban almacenados en la computadora, por ejemplo: circulares, comisiones, hojas de cálculo en formato csv, documentos de páginas web, archivos de programa en lenguajes: python, java y php; entre otros. En este caso, se hizo para determinar el potencial del algoritmo de cifrado, y verificar su viabilidad en el tratamiento de textos grandes. Los pocos experimentos realizados, en su mayoría tuvieron éxito; sin embargo, no se presenta en este trabajo información de dichos resultados, ya que, se considera, falta profundizar en ese sentido; y es por ello, que se delimitó la investigación a trabajar con cadenas de máxima longitud de 255 caracteres, y son los resultados que se exponen en la Tabla 1. En la experimentación, para cada técnica y/o método evaluados (antes mencionados), se muestra información de resultados preliminares (en Tabla 1), que presenta los mejores resultados obtenidos hasta el momento (siendo resaltados con estilo de letra negrita).

La experimentación se realizó en dos fases, una para comprobar la viabilidad del algoritmo de cifrado (y software implementado) sobre computadoras de escritorio o portátiles; la segunda fase, se realizó utilizando un dispositivo móvil, para comprobar que el código fuente escrito en Python3, y por ende, el cifrado/descifrado de datos, podría ser factible su aplicación en cómputo móvil, y de este modo, poder recomendar el uso de las nuevas propuestas de algoritmos, aquí presentadas, en ambos tipos de equipos. Cabe mencionar, que a pesar de haber utilizado el mismo código fuente de Python 3, en una computadora con Windows 7; así como, en un dispositivo móvil con Android 8.1; al final, se hizo una prueba comparativa entre los resultados y/o código fuente generados, y no se observó diferencia significativa. Los algoritmos implementados funcionaron en las mismas condiciones y con comportamientos equivalentes, en ambas plataformas.

Cada una de las muestras, generadas por el Método π (PI), se les aplicaron los algoritmos de descifrado de datos, utilizando las diversas variantes descritas en secciones previas; realizando cada experimento por separado, usando como patrones nuevos 'X', las muestra de control generadas por la validación cruzada. Si el algoritmo de cifrado logró descifrar completamente el patrón de entrenamiento cifrado, se anota un acierto o se suma 100% en ese rubro parcial; de lo contrario, se cuenta los caracteres que no fueron descifrados y se incluye a la sumatoria el n% cifrado; sino logró descifrar el mensaje se suma 0% y se cuenta como error, usando la fórmula de la precisión: Global Accuracy = Precisión = ( (Total de Aciertos * 100) / (Total de Patrones Nuevos) ). Es decir, en lugar de utilizar clasificación de patrones (1-NN) para evaluar, se utilizó el algoritmo para descifrado de datos: Si el mensaje obtenido, después del descifrado del texto, era igual a la etiqueta que corresponde a dicho patrón se consideraba un acierto. Este fue el criterio considerado para evaluar las muestras de datos cifrados. Entonces, en la Tabla 1 se muestran los promedios obtenidos de la precisión (acierto) para cada muestra de datos cifrados, que ha sido aplicado el proceso de clasificación con validación cruzada, en cada experimento. En esta investigación, en la etapa de aprendizaje del método supervisado, solamente se crea la muestra de entrenamiento, pero no se aplica ningún tipo de procesamiento o limpieza.

El propósito de utilizar GAs, combinado con métodos aleatorios, fue definir reglas y evitar que el alfabeto de cifrado/descifrado tuviera valores repetidos. En la presente investigación, se utilizó un GAs para asegurar que el alfabeto de cifrado/descifrado, esté bien "revuelto", y produzca que una segunda ejecución del cifrado de datos, sea difícil que coincida con una anterior, aún cifrando el mismo texto, deberá cada ejecución tener un alfabeto ordenado diferente, aleatoriamente. Es decir, garantiza que un mismo texto, cifrado dos veces, tenga diferente alfabeto, y por ende, distinto contenido del paquete cifrado (ver Figura 1). Lo cual se logra con la combinación de métodos: Noised Pseudo-Hexadecimal GAs + Noised Cyphered GAs Dictionary.

TABLA 1.- RESULTADOS PRELIMINARES DE LA VELOCIDAD Y ACIERTO EN CIFRADO/DESCIFRADO UTILIZANDO MUESTRAS DE TEXTO CON LONGITUD MÁXIMA DE 255 CARACTERES

| CLASIFICACIÓN | ALGORITMO DE CIFRADO | TIEMPO DE CIFRADO (CPS) | TIEMPO DE DESCIFRADO (CPS) | ACIERTO (% EN PROMEDIO) |
|---|---|---|---|---|
| **Algoritmos basados en Tablas HASH** | MD5 | 0.5 | POR DICCIONARIO | **100.00%** |
| | SHA1 | 0.5 | POR DICCIONARIO | **100.00%** |
| **Cifrado por sustitución o desplazamiento** | Cifrado del César | 0.5 | 0.5 | 95.20% |
| | Random Caesar I | 0.5 | 0.5 | *85.12%* |
| | Random Caesar II | 0.5 | 0.5 | 92.80% |
| **Cifrado con Inteligencia Artificial** | Noised Pseudo-Hexadecimal GAs + Noised Cyphered GAs Dictionary | 2.5 | 0.5 | **98.00%** |
| **PROMEDIO** | | **0.8333** | **0.3333** | **95.1866 %** |


Figura 1.- Cifrado de texto "Pseudo-Hexadecimal" (se muestra como para un mismo texto, genera un cifrado distinto en cada ejecución del algoritmo)

El texto cifrado completo en la imagen de la izquierda, de la Figura 1, es el siguiente:

c8c71f0g2f240h0U5a060Wdf0Db89fe4db5af80Pdfd89e241f0�O4a0Z960w4b0.040A0.0j0F0.0Id90ad94ff080Mff0H9cff0k7affb301ff980Ñffb4adff0R47ff0v9fff0M0vffec0Vff0z80ff590kff8d06ff0s78ff230zffacc6ff6c48ff0Ñd5ff7c30ff0q89ff0¥0hffca65ff0T04ff0�8ffbd0affe3b4ff0a0off750cff7d40ff460Xff0E0pff0l0qff720Bff0K05ff05abff090Effaa0wff310Gff2409ff0ce4ff0d0Off0m0Nffe67eff0pbeff0ndfff0r0gff0P02ff0f0Jff0A07ff0Y0Qff03ccffb10jff0e5aff5c00ff0F4cffd00Hff000Tff0X0Lffb80ñff42cdff078fff0B1fff0N0Iff0Ofaff0J0fff0V0mffaf86ff8403fff308ffef0Rff010dff7f0¥ff8899ff612effbc0Dff020iff370tffa47cff0S5fff0udaff0W5cff0Gd8fffdb6ffbf0uff770yff0y0Cff890nff0Z23ff0t77ff0Q0bffab0lffc66dff0o96ff0L0Kff0i24ff0U8eff448dff0C0Yffe50Sff292bff0bbdffbe0sff0ñ0rff4846ff4d0eff

El texto cifrado completo en la imagen de la derecha, de la Figura 1, es el siguiente:

0P0R2e0tf00q0k460s0V0Tce0h0P0Q68840s0Xddce4dbe0q0F0b410f0o300u4b0.c9440.0K0I0.05a90d108ff0b57ff0M0fff813aff0jfeffbf0Yff0W0nff6ba4ffe96aff0R87ff2c20ff1e0Vff600Cffb00hff0ñ0iffde0Jff80d9ff850Hffa30kff0p8cff45f7ff0Q7bff250Uffc7f9fff592ff900wffcc48ff020Mff0i0qff0m0uff0T0Ñff060gff0z7affc60zffb4f4ff0U0pfffda8ff860Off0N0dff72cbff0nceff040aff0l7eff790ñff000eff5d00ffe206ff0qdfff0Y0Lffa80Aff0H05ffee0Zff030rff0r24ff8d03ff0A0cfff1d3ffbb0Dff0L0Bfff369ff5609ff420tff0d41ff0S0Eff0E0Nff0Z0Iff630Gffa2c3ff0G77ff070Kff0B2eff57eeff0sc2ff0y61ffa50vffe6e2ffabedff9b0yff0e0sffd988ff0D32ff0802fff807ff0�Wff0v0Qff0o30ffaac7ff0J0�0l04ff0Ñ0¥ff460Sff5a0Fff01efffca9bff090jff0gf5ff0cc4fff090ff0C0mff0¥bfffdbb1ff0ab3ff0wfbff0O98ffcf0Xff1f01ff

En la presente investigación, para el trabajo con: Noised Pseudo-Hexadecimal GAs, solo se utiliza la Regla 1-NN como "wrapper" en el algoritmo genético; y se utilizan muestras de entrenamiento para evaluar los resultados preliminares de la experimentación. Este sistema de reconocimiento de patrones usado como "wrapper" (o clasificador 1-NN) permite evaluar la población y seleccionar los cromosomas más aptos que formarán la siguiente generación del alfabeto de cifrado. El uso de la Regla 1-NN no incluye variantes para generar el alfabeto de cifrado; tampoco utiliza aprendizaje continuo, solamente se trabajó

con Aprendizaje Supervisado, en la etapa de experimentación, clasificando los textos cifrados/descifrados, formando patrones dentro de una muestra de entrenamiento, cuya etiqueta de clase corresponde al mensaje o texto original. También, se utilizó Aprendizaje No Supervisado, en la parte del algoritmo genético, al momento de clasificar los cromosomas con un sistema de reconocimiento de patrones, para poder decidir cuáles fueron considerados más aptos y formar la siguiente generación (es decir, al clasificar con distancia euclidiana se tuvo que agrupar parcialmente, evaluando dos clases: de cromosomas más parecidos al alfabeto de una población inicial, que computaron la menor distancia y los cromosomas menos parecidos entre sí), que en términos generales, sobresalieron aquellos que no tenían caracteres ASCII repetidos después de la mutación; así como, aquellos que fueron observados con mayor distancia; es decir, seleccionando lo contrario reportado por la Regla NN (Cover T. M. & Hart P. E. , 1967), reitero, menos parecidos entre sí, después de haber evaluado con la distancia euclidiana o inversa de la Regla 1-NN. Es por ello, que todas muestras de entrenamiento guardaban mensajes de texto cifrados (en valor numérico entero y hexadecimal correspondiente a caracteres de la tabla ASCII), y en lugar de hacer uso de la Regla 1-NN como clasificador, fueron sometidos al algoritmo de descifrado; y el resultado generado por el mismo, se comparó con la etiqueta (que tenía el mensaje de texto original), marcando error si no coinciden y anotando acierto en caso de ser el mismo valor. Cabe mencionar, que no se realizaron experimentos con ME desbalanceadas, y tampoco, se hizo la aplicación de ningún método de preprocesamiento de las muestras de entrenamiento. Sin embargo, **sí** se detectaron casos atípicos o ruidosos en los experimentos realizados con el algoritmo: Random Caesar I. Se descubrió que, al utilizar 255 caracteres del ASCII, producía mucho error en el descifrado de datos, sobretodo en aquellos caracteres o valores ASCII que no son imprimibles en pantalla, tampoco fueron legibles en almacenamiento (en archivos formato TXT con codificación ANSI o UTF-8), así como, en salida del área o campo de texto de la aplicación implementada en lenguaje Python3, lo ocasionó la reproducción de los errores.

## 5.- Conclusiones y Trabajos Futuros

### 5.1.- Conclusiones

De acuerdo con los resultados obtenidos durante la experimentación, podemos apreciar que: **(a)** Los algoritmos o métodos de cifrado tradicionales (por desplazamiento), estudiados en esta investigacón, tal como es el caso del Cifrado del César o Algoritmo Caesar (Luciano & Prichett, 1987 ; Gómez et.al, 2012 ; Stinson D.R., 2005 ; Barranco & Galindo, 2022); así como, los algoritmos basados en Tablas Hash, me refiero a: MD5 (Willa, 2018 ; Lake, 2023) y SHA1 (Lowery & Pereyra, 2023 ; PortalCripto, 2023); se observó que cuando se usa el modelo estándar, que **no** incrusta contraseña o claves públicas; a pesar de ser veloces en el cifrado/descifrado de datos, y aunque generan un documento cifrado de tamaño igual que la cadena de mensaje original, resultan vulnerables porque no presentan variaciones en el contenido para una misma cadena de texto. Por ende, podría ser descifrado usando un diccionario con uso de procesamiento de lenguaje natural con IA. **(b)** Las propuestas: Random Caesar I y Random Caesar II, a pesar de ser más veloces y que generan un documento cifrado de tamaño menor que Noised Pseudo-Hexadecimal GAs; se observó que no son tan rápidos como los estándares basados en desplazamiento (Caesar) y/o Tablas Hash (MD5 y SHa1). Además, los algoritmos Random Caesar, en ocasiones, presentan problemas en el descifrado de datos, sobretodo, cuando algún caracter aleatorio seleccionado, resulta ser no imprimible en pantalla o decodificado con pérdida en formatos: ANSI o UTF-8. **(c)** La propuesta: Noised Pseudo-Hexadecimal GAs, a pesar que no es muy rápida para el cifrado de datos, porque se genera en el procesamiento de cifrado, un archivo de tamaño muy extenso; la operación de descifrado es viable, porque consume menos tiempo que el cifrado de datos con esa misma metodología. Además, el contenido del mensaje cifrado, para una misma cadena de texto, difiere en cada ejecución del algoritmo; y debido a que, se tiene inyectado ruido en el mensaje cifrado, por el concepto: “Pseudo-Hexadecimal”, resulta menos vulnerable a ataques, en comparación con el resto de los métodos o algoritmos, evaluados en esta investigación. **(d)** En términos generales, todos los algoritmos, técnicas y/o métodos evaluados(as) en esta investigación, reportan resultados de acierto en

el cifrado/descifrado de datos, muy semejantes; cuya diferencia de porcentajes, en promedio general, difieren aproximadamente entre 2% y 14.88%. **(e)** Finalmente, la propuesta: Noised Pseudo-Hexadecimal GAs, fue evaluada con software especializado para descifrado de datos, resultados que no se presentan en esta investigación, ya que, aún falta concluir varios experimentos. Sin embargo, al momento, no se ha podido descifrar el paquete generado por: Noised Pseudo-Hexadecimal GAs, ya que, dichas herramientas de software especializadas para el descifrado de datos, solicitan información, tal como: clave pública o privada, tipo de algoritmo empleado, entre otros datos; los cuales **no** contiene el paquete generado por la nueva propuesta de cifrado basada en Pseudo-Hexadecimal. **(f)** Por último, en términos generales, todas las técnicas y/o métodos evaluados(as) en esta investigación, reportan resultados en velocidad de cifrado/descifrado, muy equivalentes, que varian entre 0% y 1% de diferencia de cps (medida en ciclos por segundo).

### 5.2.- Trabajos Futuros

Deacuerdo con lo observado en las fases de experimentación, se puede apreciar que el Cifrado del Caesar (así como, variantes estudiadas) obtiene resultados distintos en la evaluación cuando se sustituye el criterio del desplazamiento estático (K), por uno dinámico (K[i]), generado de manera aleatoria (con reemplazo). En caso del algoritmo: Noised Pseudo-Hexadecimal GAs, se utilizó la distancia euclideana para el cálculo de la Regla 1-NN. Sin embargo, en algunas investigaciones (Wilson & Martinez, 1997; Wilson & Martinez, 1998; Kittipong et. al. , 2015) se exponen una serie de métricas para realizar cálculos de distancias; y partiendo de esas bases, se puede generar un trabajo futuro, consistente en repetir la aplicación de todos los experimentos utilizando variantes de los métodos y/o técnicas descritos(as) en el presente trabajo; diseñando nuevas propuestas basadas en los criterios de distancia expuestos por: Kittipong et. al. (2015) y/o Wilson & Martínez (1997, 1998); todo con la finalidad de observar qué función de distancia obtiene mejores resultados en la clasificación de patrones, cuando se utilizan variantes de la Regla 1-NN, en el algoritmo: Noised Pseudo-Hexadecimal GAs. Otro trabajo futuro, puede ser considerado, sustituyendo la Regla 1-NN por otras variantes; por ejemplo, utilizar Regla k-NN o k-NCN (Chaudhuri, 1996; Rangel, 2021; Rangel, 2002; Moreno-Seco et. al. , 2001; Sánchez et. al. , 2000 ; Shankar & Karypis 2000 ; Eui-Hong & Karypis, 1999; Sánchez, 1998 ; Sánchez et. al. , 1997). Un trabajo adicional, podría ser, modificar los algoritmos aquí presentados, para incorporar una clave pública, o contraseña de cifrado, y poder comparar resultados con variantes del SHA1, que utilizan dicho procedimiento. Del mismo modo, se podría extender la investigación, evaluando más algoritmos populares, tales como: Polybios, Vigenére, RSA, AES, DES, por mencionar algunos. .

## Referencias Bibliográficas

Aha, D. W. ; Dennis K. ; Marc, K.A (1991). Instance-Based Learning Algorithms. Machine Learning, 6, pp. 37-66.

Albentia Systems. (2011). Seguridad En Redes WiMAX. Disponible en: http://www.albentia.com/Docs/WP/ALB-W-000006spA4_Seguridad%20en%20redes%20WiMAX.pdf.

Álvarez, D. (2019). Algunos Aspectos Jurídicos Del Cifrado De Comunicaciones. Derecho PUCP, núm. 83, 2019, pp. 241-264. Pontificia Universidad Católica del Perú. DOI: https://doi.org/10.18800/derechopucp.201902.008. Disponible en: http://www.redalyc.org/articulo.oa?id=533662765008.

Barandela, R. (1990). La Regla NN con muestras de entrenamiento no balanceadas. Investigación Operacional, X (1), 45-56.

Barandela R. ; Cortés N. ; A. Palacios (2001a). "The Nearest Neighbor rule and the reduction of the training sample size". IX Symposium of the Spanish Society for Pattern Recognition, Castellon, Spain.

Barandela, R. ; Gasca, E. (2000). "Decontamination of training samples for supervised pattern recognition methods". In: Advances in Pattern recognition, Lecture Notes in Computer Science, vol. 1876, F. Ferri et al. (eds.), Springer, Berlin, 621-630.

Barandela, R. ; Juarez M. (2001). "Ongoing Learning for Supervised Pattern Recognition". Submitted to SIBGRAPI-2001, Brazil.

Barandela, R. ; Najera T. (2002). Supervised Pattern Recognition With Incomplete Training Samples. Enviado a: Workshop on Statistical Pattern Recognition, Canadá.

Barandela, R. ; Rangel, E. ; Sánchez, JS. ; Ferri, FJ. (2003). Restricted Decontamination for the Imbalanced Training Sample Problem. 8th Iberoamerican Conference on Pattern Recognition , Havana (Cuba), Progress in Pattern Recognition, Speech and Image Analysis, Lecture Notes in Computer Science 2905 , A. Sanfeliu and J. Ruíz-Shulcloper (eds.), Springer-Verlag, pp. 424-431, ISBN 3-540-20590-X.

Barandela, R. ; Sánchez J. S. ; García, V. ; Rangel, E. (2001b).“Fusion of techniques for handling the imbalanced training sample problem”. In: Procedings of 6th Ibero-American Symposium on Pattern Recognition, Brasil, 2001, 31-40.

Barranco, F. ; Galindo C. (2022). Criptografía básica y algunas aplicaciones. Universidad Jaume I, Departamento de Matemáticas, Castellón, España.

Bruzzone, L. ; Serpico, S.B. (1997). Classification of Imbalanced remote-sensing data by neural networks. Elsevier Science B.V. , 0167-8655, 97. PH S0167-8655 (97) 00109-8.

Chaudhuri, B. B. (1996). A New definition of neighborhood of a point in multi-dimensional space. Pattern Recognition Letters 17, 11-17.

Chong F. ; Bib L. ; Yu S.M. ; Xiao L. ; Jun J. (2011). A Novel Chaos-based Bit level Permutation Scheme For Digital Imagen Encryption. Optics communications, volumen 284, august [en línea]. Disponible en: www.elsevier.com/locate/optcom.

Clark, A. (1994). Modern optimisation algorithms for cryptanalysis. In Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems, November 29 December 2, (pp. 258-262).

Courtois N. T. (2012). Security Evaluation Of GOST 28147-89 In View Of International Standardisation. Cryptologia, Vol. 36, no. 1, 2012, pp. 2-13. DOI 10.1080/01611194.2011.632807.

Cover, T. M. ; Hart, P. E. (1967). “Nearest Neighbor Pattern Classification”. IEEE Transactions on Information Theory, Volume IT-13, January, pages 21-27.

Dasarathy B.V., Sánchez J.S., and Townsend S. (2000). Nearest Neighbor Editing and Condensing tools - synergy exploitation”. Pattern Analysis & Applications. Vol. 3, no. 1, pp. 19-30.

Delman, B. (2004). Genetic Algorithms in Cryptography. Thesis for the Degree of Master of Science in Computer Engineering. Rochester Institute of Technology (RIT Scholar Works). Department of Computer Engineering.

Devijver, P. A. ; Kittler, J. (1980). On the Edited Nearest Neighbor Rule. From: Proceedings of the 5th International Conference on Pattern Recognition, December, pages 72-80.

Dudani, S. A. (1976). The distance-weighted k-nearest neighbor rule. IEEE TransactionsSystems Man and Cybernetics, Vol. SMC-6(4), pp. 325-327.

ELENA DATABASE (2002). Anonymous ftp at ftp.dice.ucl.ac.be, directory pub/neural-nets/ELENA/databases.

Eui-Hong (Sam) ; Karypis George (1999). Centroid-Based Document Classification: Analysis & Experimental Results. Eui-Hong (Sam) Han. From: umn.edu/general/Reports/2000 Home: E. Han HPSearch.

Fix, E. ; Hodges, J.L. Jr. (1952). “Discrimination analysis: small sample performance”. USAF School of Aviation Medicine, Randolph, Field, Tex. Project 21-49-004, Rept. 11, August.

Frank A. ; Asuncion, A. (2010). UCI machine learning repository, <http://archive.ics.uci.edu/ml/>.Irvine, CA: University of California, School of Information and Computer Science, 2010.

Freda A. (2022). "¿Qué es el algoritmo de hashing MD5 y cómo funciona?". Avast Academy. Recuperado: 30-11-2023. URL: https://www.avast.com/es-es/c-md5-hashing-algorithm#:~:text=El%20MD5%20%28algoritmo%20de%20resumen%20de%20mensajes%29%20es,persona%20a%20la%20que%20se%20lo%20ha%20enviado

Fulgueira, M. ; Hernández, O.A. ; Henry, V. (2015). Paralelización Del Algoritmo Criptográfico GOST Empleando El Paradigma De Memoria Compartida. Lámpsakos, núm. 14, pp. 18-24. Fundación Universitaria Luis Amigó Medellín, Colombia. E-ISSN: 2145-4086; julio-diciembre. DOI: http://dx.doi.org/10.21501/21454086.1633. Disponible en: http://www.redalyc.org/articulo.oa?id=613965326004.

Galende, J. (2000). Sistemas Criptográficos Empleados En Hispanoamérica. Revista Complutense de Historia de América, 26, 57-71.

Gates, G. W. (1972). The reduced nearest neighbor rule. IEEE Trans. Inform. Theory (Corresp.) Vol. IT-18, pp. 431-433, May.

Gongde Guo, Hui Wang, David Bell, Yaxin Bi and Kieran Greer (2004). KNN Model-Based Approach in Classification. ResearchGate. URL: https://www.researchgate.net/publication/2948052.

Goodman J. (1968), Introduction To Fourier Optics. New York: McGraw-Hill.

Gómez, S. ; Arias, J.D. ; Agudelo, D. (2012). Cripto-Análisis Sobre Métodos Clásicos De Cifrado.Scientia Et Technica, vol. XVII, núm. 50, abril, pp. 97-102. Universidad Tecnológica de Pereira Pereira, Colombia. ISSN 0122-1701 97. Disponible en: http://www.redalyc.org/articulo.oa?id=84923878015.

Granados, G. (2006). Introducción A La Criptografía. Revista Digital Universitaria, 7(7), s.p. Recuperado de:http://www.revista.unam.mx/vol.7/num7/art55/int55.htm.

Grundlingh, W. ; Van Vuuren, J. H. (2002). Using Genetic Algorithms to Break a Simple Cryptographic Cipher. Retrieved March 31, 2003 from http://dip.sun.ac.za/~vuuren/abstracts/abstr_genetic.htm.

Hand D., Mannila H., Smyth P. (2001). Principles of Data Mining. The MIT Press.

Hart, P.E. (1968). "The Condensed Nearest Neighbor Rule", IEEE Transactions on Information Theory, 6,4,515-516, Vol. IT-14, No. 3, May.

Hebert, S. (s.f.). A Brief History of Cryptography. Disponible en: http://cybercrimes.net/aindex.html.

Hong-Tao ; Lam , S.W. ; Hull, J.J. ; Srihari, S.N. (1993). The Design of Nearest-Neighbor Classifier and Its Use for Japanese Character Recognition. Eui-Hong (Sam) Han. From: umn.edu/general/Reports/2000 Home: E. Han HPSearch.

Hossam E.A. ; Hamdy K. ; Osama S.F.A. (2007). An Efficient CHAOS-BASED FEEDBACK STREAM CIPHER (ECBFSC) For Image Encryption And Decryption. Informática, volumen 3, pp. 121-129.

Javidi, B. ; Zhang, G. S. ; Li, J. (1997). Encrypted Optical Memory Using Double-random Phase Encoding. Appl. Opt. 36, 1054-1058.

Jiménez, M. ; Flores, O. ; González, M.G. (2015). Sistema para codificar información implementando varias órbitas caóticas. Ingeniería. Investigación y Tecnología, vol. XVI, núm. 3, julio-septiembre, pp. 335-343. ISSN 1405-7743 FI-UNAM / ISSN: 1405-7743. Universidad Nacional Autónoma de México. Distrito Federal, México. Disponible en: http://www.redalyc.org/articulo.oa?id=40440683002.

Johns, M. V. (1961). An empirical Bayes approach to non-parametric two-way classification, In Studies in Item Analysis and Prediction, H. Solomon, Ed Stanford, Calif,: Standford University Press.

Kalsi, S. ; Kaur, H. ; Chang, V. (2018). DNA Cryptography and Deep Learning using Genetic Algorithm with NW algorithm for Key Generation. Convergence of Deep Machine Learning and Nature Inspired Computing Paradigms for Medical Informatics. Image & Signal Processing; In Journal of Medical Systems, volume 42, Article number: 17. DOI: https://doi.org/10.1007/s10916-017-0851.

Kanal, L. N. (1963). Statical methods for pattern classification. Philco Rept, originally appeared in T. Harley et al., Semi-automatic imagery screening research study and experimental investigation, Philco Reports, V043-2 and v043-3, Vol. I, sec. 6 and Appendix H, prepared for U.S. Army Electronics Research and Development Lab. Under Contract DA-36-039-sc-90742, March 29.

Kaushik Chakrabarti, Kriengkrai Porkaew and Sharad Mehrotra. (2000). Efficient Query Refinement in Multimedia Databases. Eui-Hong (Sam) Han. From: umn.edu/general/Reports/2000 , Home: E. Han HPSearch.

Khaled El-Maleh, Mark Klein, Grace Petrucci and Peter Kabal. (2000). Speech / Music Discrimination for Multimedia Applications. Eui-Hong (Sam) Han. From: umn.edu/general/Reports/2000 Home1: http://WWW.TSP.ECE.McGill.CA. Home2: E. Han HPSearch.

KeepCoding (2023). ¿Qué es SHA-1?. Keepcoding. Recuperado el 02-02-2024. URL: https://keepcoding.io/blog/que-es-sha-1/

Khan, D. (1967). The Codebreakers. Macmillan Publishing Company, New York.

Kittipong Chomboon, Pasapitch Chujai, Pongsakorn Teerarassamee, Kittisak Kerdprasop, Nittaya Kerdprasop (2015). An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm. Proceedings of the 3rd International Conference on Industrial Application Engineering 2015. The Institute of Industrial Applications Engineers, Japan. DOI:10.12792/iciae2015.051.

Koplowitz, J. ; Brown, T A. (1978). On the relation of performance to editing in nearest neighbor rules. In: Proceedings of the 4th International Joint Conference on Pattern Recognition, Japan.

Kubat, M. ; Holte, R. ; Matwin, S. (1997). Learning when Negative Examples Abound. Proceeding of the 9th European Conference on Machine Learning, ECML´97, Prague.

Kubat, M. ; Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. Proceedings of the 14th International Conference on Machine Learning, 179-186.

Kuo Yin-Hung ; Man-Hon Wong. (2000). Web Document Classification based on Hyperlinks and Document Semantics. A.-H. Tan and P. Yu (Eds): PRICAI 2000 Workshop on Text and Web Mining Melbourne, pp. 44-51, August.

Kuncheva, L. I. ; Jain L. C. (1999). "Nearest Neighbor Classifier: Simultaneous editing and feature selection". Pattern Recognition Letters, 20, 1149-1156.

Lake, J. (2023). The MD5 algorithm (with examples). Comparitech. Recuperado el 02-02-2024. URL: https://www.comparitech.com/blog/information-security/md5-algorithm-with-examples/

Lewis, D. ; Catlett, J. (1994). Heterogeneous Uncertainty Sampling for Supervised Learning. Proceedings of the 11th International Conference on Machine Learning, ICML´94 (pp. 148-156), New Brunswick, New Jersey, Morgan Kaufmann.

Linfei, C. ; Daomu, Z. (2005). Optical Image Encryption Based On Fractional Wavelet Transform, Opt. Comm. Vol. 254 (2005) p.p. 361-367.

Loftsgaarden, D.O. and C.P. Quesenberry (1965). A nonparametric estimate of a multivariable density function, Annals Math Stat., vol. 36, pp. 1049-1051, June.

Lowery J. (2023). "MD5 vs SHA-1 vs SHA-2: ¿Cuál es el Hash cifrado más seguro y cómo verificarlos?". FreeCodeCamp. Recuperado: 30-11-2023. URL: https://www.freecodecamp.org/espanol/news/md5-vs-sha-1-vs-sha-2-cual-es-el-hash-cifrado-mas-seguro-y-como-verificarlos/

Lowery, J ; Pereyra, E. (2023). MD5 vs SHA-1 vs SHA-2: ¿Cuál es el Hash cifrado más seguro y cómo verificarlos?. FreeCodeCamp. Recuperado el 02-02-2024. URL: https://www.freecodecamp.org/espanol/news/md5-vs-sha-1-vs-sha-2-cual-es-el-hash-cifrado-mas-seguro-y-como-verificarlos/

Luciano; D. ; Prichett, G. (1987). Cryptology: From Caesar Ciphers To Public-key Cryptosystems. The College Mathematics Journal, vol 18 pp 2-17.

Matoba, O. ; et al. (2000). Secure Optical Storage Using Fully Phase Encryption, Journal of Applied Optics,

vol. 39, pp. 6689-6694, December 10.

Matthews, R.A.J. (1993). The use of genetic algorithms in cryptanalysis. Cryptologia, 17(4), 187-201.

Mendoza, J.C. (2008). Demostración De Cifrado Simetrico Y Asimetrico. Ingenius. Revista de Ciencia y Tecnología, núm. 3, pp. 46-53. Universidad Politécnica Salesian. Cuenca, Ecuador. ISSN: 1390-650X. Disponible en: http://www.redalyc.org/articulo.oa?id=505554806007.

Menezes, A.J. ; Van Oorschot, P.C. ; Vanstone, S.A. (1997). Handbook of applied cryptography.

Merz C.J. ; Murphy P.M. (1998). UCI Repository of Machine Learning Databases, University of California at Irvine, Departament of Information and Computer Science. http://www.csi.uci.edu/ ~mlearn (updated as: https://archive.ics.uci.edu/ml/datasets.php).

Mitchell, M. and Holland, J. H.(1993). When Will a Genetic Algorithm Outperformance Hill-Climbing ". Technical report, Santa Fe Intitute.

Mogensen, P. ; Glückstad, J.A. (2000). Phase-based Optical Encryption System With Polarisation Encoding. Opt. Commun. pp. 173, 177-183

Mogensen, P. ; Glückstad, J. (2001) Phase-only optical decryption of a fixed mask. Appl. Opt. 40,1226-1235.

Mogensen, P. ; Glückstad, J. (2000). Phase-only Optical Encryption. Opt. Lett. 25, 566-568.

Mogensen, R. E. ; Glückstad, J. (2001). High Capacity Optical Encryption System Using Ferro-Electric Spatial Light Modulators. J. Optics A. 3, 10-15.

Moreno-Seco F. ; Iñesta, J.M. ; Micó, L. ; Oncina, J. (2001). "Fast k-Neighbor classification of human vertebrae levels". In Proceedings of the IX Symposium on Pattern Recognition and Image Analysis Benicasim, Castellón (Spain), 16-18 May.

Nagaraj, S. ; Srinadth, V. (2015). Data Encryption and Authetication Using Public Key Approach. International Conference on Computer, Communication and Convergence (ICCC 2015); DOI: 10.1016/j.procs.2015.04.161/ScienceDirect. Procedia Computer Science 48 ( 2015 ) 126 - 132, 1877-050, Elsevier B.V. Interscience Institute of Management and Technology, Bhubaneswar, Odisha, India. Disponible en: www.sciencedirect.com.

Nash, A. (2004). Infraestructura De Claves Públicas, Editorial McGraw Hill,México.

Nils-Nilson (1965). Learning Machines. New York: McGraw Hill, pp 120-121.

Nomura, T.; Javidi, B. (2000). Optical Encryption System Using A Binary Key Code, Journal of Applied Optics, vol. 39, pp. 4783-4787, September 10,

Oppliger, R (s.f). Contemporary cryptography, 1ra. ed., Boston, Artech.

Pieprzyk, J. ; Tombak, L. (1994). Soviet Encryption Algorithm. University of Wollongong, Department of Computing Science.

Pisarchik, A.N. ; Flores-Carmona N.J. (2006). Computer Algorithms For Direct Encryption And Decryption Of Digital Images For Secure Communication, Proceeding of the 6th WSEAS international conference on applied computer science, (Canary Islands, Spain), pp. 29-34.

Pisarchik, A.N. ; Zanin, M. (2008). Imagen Encryption Witch Chaotically Coupled Chaotic Maps. Elsevier Physica, D 237, abril [en línea]. Disponible en: www.elsevier.com/locate/physd.

PortalCripto (2023). Algoritmo Hash SHA-256: qué es y cómo funciona. PortalCripto. Recuperado el 02-02-2024. URL: https://portalcripto.com.br/es/DICIONARIO/Algoritmo-hash-sha-256-qu%C3%A9-es-y-c%C3%B3mo-funciona/

Progress Software Corporation; Telerik (2020-2022). Cifrado Y Transferencia De Archivos: Los Mejores Cifrados Seguros Para La Transferencia De Archivos. Ipswitch Blogs. Recuperado de: https://ipswitch.com/amp/es/los-mejores-cifrados-seguros-para-la-transferencia-de-archivos/

Rajan, B. ; Saumitr, P.A. (2006). Novel Compression And Encryption Scheme Using Variable Model Arithmetic Coding And Coupled Chaotic System. IEEE Transactions on circuits and system- I, volumen 53 (número 4), abril.

Rangel, E. (2002). "Vecinos Envolventes para Variantes de la Regla del Vecino más Cercano". Ph.D. Thesis, Instituto Tecnológico de Toluca, México. ["Variants For Nearest Centroid Neighbour"].

Rangel, E. ; García, O. (2002). Nearest Centroid Neighbour, An Alternative in the Speech Recognition for the Execution from a Mobile Robot Simulator (Applied to the Imbalanced Training Sample Problem). Publicado In: 9th International Congress On Computer Science Research (CIICC 02) . October 23-25 2002 - Puebla, México. ISBN: ISBN-970-18-8526-0 (Artículo En Extenso)

Rangel E. ; Barandela R. (2004). Nearest Centroid Neighbour, An Alternative in Pattern Recognition for Detecting New Tasks in a Mobile Robot Simulator. Enviado a: 11th International Congress On Computer Science Research (CIICC04). September 31, October 1, 2. Ciudad de México - México (Artículo En Extenso).

Rangel, E. ; Rodríguez, C. (2018). "Un Estudio Con Variantes De La Regla NN, Como Alternativa En Inteligencia Artificial Para Incrementar La Precisión En Clasificación De Patrones". Publicado En: Primer Congreso Nacional De Investigación En Ciencia E Innovación De Tecnologías Productivas. Tecnológico Nacional De México (campus: Instituto Tecnológico de Cd. Altamirano). Noviembre, 2018. Cd. Altamirano, Estado De Guerrero, México. (Artículo En Extenso).

Rangel, E. (2019). Resultados Preliminares Con Variantes De La Regla NN, Como Alternativa En Inteligencia Artificial, Para Clasificación Usando Muestras De Entrenamiento Desbalanceadas. Publicado En: Segundo Congreso Nacional De Investigación En Ciencia E Innovación De Tecnologías Productivas. Tecnológico Nacional De México (campus: Instituto Tecnológico de Cd. Altamirano). Noviembre, 2019. Cd. Altamirano, Estado De Guerrero, México. (Artículo En Extenso).

Rangel, E. (2020). La Regla NN Como Alternativa Aplicable A La Agricultura 5.0. Enviado a: Tercer Congreso Nacional De Investigación En Ciencia E Innovación De Tecnologías Productivas. Tecnológico Nacional De México (campus: Instituto Tecnológico de Cd. Altamirano). Noviembre, 2020. Cd. Altamirano, Estado De Guerrero, México. (Artículo En Extenso).

Rangel, E. (2021). Vecinos Envolventes Como Alternativa En Inteligencia Artificial Para Incrementar La Precisión En Clasificación De Patrones. Publicado En: Cuarto Congreso Nacional De Investigación En Ciencia E Innovación De Tecnologías Productivas. Tecnológico Nacional De México (campus: Instituto Tecnológico de Cd. Altamirano). Noviembre, 2021. Cd. Altamirano, Estado De Guerrero, México (Artículo En Extenso).

Rangel, E. (2022). La Regla De Los k Vecinos Más Cercanos (k-NN) Basada En Distancia De Manhattan (City-Block) Para Mejorar La Clasificación De Patrones. Publicado En: Quinto Congreso Nacional De Investigación En Ciencia E Innovación De Tecnologías Productivas. Tecnológico Nacional De México (campus: Instituto Tecnológico de Cd. Altamirano). Noviembre, 2022. Cd. Altamirano, Estado De Guerrero, México (Artículo En Extenso).

Reddaiah, B. (2016). A Study on Pairing Functions for Cryptography. IJCA (0975-8887), Vol. 149, No. 10, September, pp.4-7.

Reddaiah, B. (2019). A Study on Genetic Algorithms for Cryptography. International Journal of Computer Applications (0975 – 8887). Volume 177 - No. 28, December. Department of Computer Applications. Yogi Vemana University Kadapa, A.P, India.

Ritter, G.L. ; Woodruff, H. B. ; Lowry, S. R. ; Isenhour, T. L. (1975). An Algorithm for a Selective Nearest Neighbor Decision Rule. From: IEEE Transactions on Information Theory, Volume IT-21, Number 6, November, pages 665-669.

Rodríguez, J. (2020). Operadores Genéticos Aplicados A La Criptografía Simétrica. Proyecto De Grado. Universidad Distrital Francisco José De Caldas. Facultad De Ingeniería. Ingeniería De Sistemas. Bogotá, Colombia.

Rosen, J. ; Javidi, B. (2001). Optical Encryption Using Embedded Images, Journal of Applied Optics, accepted for publications, to appear.

Ross-Quinlan ; J. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

Rueda, A. S. ; Lasprilla, M. (2002). Encriptación Por Conjugación De Fase En Un BSO Utilizando Señales Ópticas De Baja Potencia, Rev. Col. Fís., Vol. 34, No.2, (2002), P.P.636-640.

Rueda, J.E. ; Romero, A. L. ; Castro, L.M. (2005). Criptografía Digital Basada En Tecnología Óptica. Bistua: Revista de la Facultad de Ciencias Básicas, vol. 3, núm. 2, julio, pp. 19-25. ISSN 0120 - 4211. Universidad de Pamplona, Colombia. Disponible en: http://www.redalyc.org/articulo.oa?id=90330203.

Sanchéz, C. (2014). Teoría De La Información Y Teoría De Códigos. Disponible en: http://www.cesans.net/cripto/DES_Diferencial-Carlos_Sanchez.pdf.

Sánchez, J. S. (1998). Aprendizaje y clasificación basados en criterios de vecindad. Métodos alternativos y análisis comparativo. In Spanish, Ph. D. Thesis. Universitat Jaume I, Departament d´Informática, Castelló, Spain. January.

Sánchez, J. S. ; Barandela, R. ; Marques, A.I. ; Alejo, R. ; Badenas, J. (2001). Análysis of new techniques to obtain quality training sets. Preprint submitted to Elsevier Science 29 October.

Sánchez, J. S. ; Pla F. ; Ferri FJ. (1997a). Using the Nearest Centroid Neighbourhood concept for editing purposes. 7th., Spanish Symposium on Pattern Recognition and Image Analysis, pp. 175-180, Barcelona (Spain), ISBN 84-922529-0-1.

Sánchez, J.S. ; Pla, F. ; Ferri, F.J. (1997b). Prototype selection for the nearest neighbor rule through proximity graphs. Pattern Recognition Letters 18, 507-513.

Sánchez, J. S. ; Pla F. ; Ferri FJ. (2000). Adaptative learning from nearest Centroid neighbours for the nearest neighbour rule. Pattern Recognition and Applications, Frontiers in Artificial Intelligence and Applications 56, M.I. Torres and A. Sanfeliu (Eds), IOS Press, pp. 29-36, ISBN 1-58603-034-5.

Sebas, C. (2023). ¿Qué son los Algoritmos Genéticos en las Inteligencias Artificiales?. Manuales y Tutoriales de Informatica. Recuperado de: https://aprendeinformaticas.com/ia/

Sebestyen, G. (1962). Decision Making Process in Pattern Recognition. New York: Macmillan, pp. 90-92.

Shankar-Shrikanth; Karypis, G. (2000). Weight-Adjustament Schemes for a Centroid-Based Classifier. Eui-Hong (Sam) Han. From: umn.edu/general/Reports/2000 Home: E. Han HPSearch.

Singh, S. (2000). Los códigos secretos. Madrid: Debate.

Sinha, A. ; Singh, K. (2003). A Technique For Image Encryption Using Digital Signature, Optics Communications, 218, p.p.229-234, (2003).

Skalak, D. B. (1994). "Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms". In: Proceedings of the Eleventh International Conference on Machine Learning (ML94). Morgan Kaufmann, pp. 293-301.

Solís, F. ; Pinto, D. ; Solís, S. (2017). Seguridad De La Información En El Intercambio De Datos Entre Dispositivos Móviles Con Sistema Android Utilizando El Método De Encriptación RSA. Enfoque UTE, V.7-Sup.1, Feb.2017, pp.160 - 171. e-ISSN: 1390-6542 / p-ISSN 1390-9363. Disponible en: http://ingenieria.ute.edu.ec/enfoqueute/

Stinson D.R. (2005). Cryptography: theory and practice. Chapmany Hall/CRC, 2005.

Swonger, C. W. (1972). Sample Set Condensation for a Condensed Nearest Neighbor decision rule for pattern recognition. By C.W. Swonger from Frontiers of Pattern Recognition, edited by S. Watanabe, pages 511-519. Copyright (c) by Academic Press, Inc., reprinted with permission.

Tajahuerce, E. ; Javidi, B. (2000). Encrypting Three Dimensional Information With Digital Holography, Journal of Applied Optics, vol. 39, pp. 6595-6601, December 10.

Tajahuerce, E. ; et al. (2001). Optical Security And Encryption With Totally Incoherent Light, Journal of Optics Letters, vol. 26, pp. 678-681, May 15.

Tajahuerce, E. ; et al. (2000). Optoelectronic Information Encryption Using Phase-shift Interferometry, Journal of Applied Optics, Vol. 39, pp. 2313-2320, May 10.

Tan, H.C.A. (2005). Encyclopedia Of Cryptography And Security. pp 115.

Tanenbaum A. S. (2003). Modern Operating Systems. Prentice Hall, 2003.

Thakur, J. ; Kumar, N. (2011). DES,u AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. International Journal of Emerging Technology and Advanced Engineering, 6-12.

Tomek I. (1976). Two modifications of CNN, IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-6, Nov., pp. 769-772.

Tomek, I. (1976b). An Experiment with the Edited Nearest-Neighbor Rule. From: IEEE Transactions on Systems, Man, and Cybernetics, Volume SMC-6, Number 6, June, pages 448-452.

Toussaint, G. T. (1992). A Counter-Example to Tomeks Consistency Theorem for a Condensed Nearest Neighbor Decision Rule. Eui-Hong (Sam) Han. From: umn.edu/general/Reports 2000 Home: E. Han HPSearch.

Tzanetakis George ; Perry Cook. (2000). Sound Analysis Using MPEG Compressed Audio. Eui-Hong (Sam) Han. From: umn.edu/general/Reports/2000 Home: E. Han HPSearch.

Van, H. C. ; Jajodia, S. (2011). Encyclopedia Of Cryptography And Security. Springer Science & Business Media, 2011. 1416p. ISBN:978-14419-5907-2.

Vander-Lugt, A. (1964). Signal Detection By Complex Spatial Filtering. IEEE transactions on Information, Vol. 10, (1964), pp. 139.

Vargas, L. (2010). ProyBach. Escuela de Ingeniería Eléctrica de Costa Rica. [Online]. Disponible en: http://eie.ucr.ac.cr/uploads/file/proybach/pb2009/pb2009_036.pdf.

Wang, H. (2002). Nearest Neighbours without k: A Classification Formalism based on Probability, technical report, Faculty of Informatics, University of Ulster, N. Ireland, UK.

Willa Dhany, Hanna ; Fahmi Izhari; Hasanul Fahmi; Mr Tulus; Mr Sutarman (2018). Encryption and Decryption using Password Based Encryption, MD5, and DES. Published by Atlantis Press. ISSN: 2352-5398. Open Access: CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

William, S. (1999). Cryptography and Network Security: Principles and Practice, 2nd edition, Prentice-Hall, Inc., pp 23-50.

Wilson, D.L. (1972). Asymptotic Properties of Nearest Neighbor Rules using Edited Data, IEEE Transactions on Systems, Man and Cybernetics 2(3) , 408-421.

Wilson, D. Randall ; Martinez, Tony R. (1997). Instance Pruning Techniques. In: Fisher, D., ed., Machine Learning: Proceedings of the Fourteenth International Conference (ICML´97), Morgan Kaufmann Publishers, San Francisco, CA, pp. 404-411.

Wilson, D.R. ; Martinez, T. R. (1998). Reduction Techniques for Exemplar-Based Learning Algoritms. To appear in Machine Learning. Eui-Hong (Sam) Han. From: umn.edu/general/Reports/2000 Home: E. Han HPSearch.

Wilson, D.R. ; Martinez, T. R. (2000). Reduction Techniques for Instance-Based Learning Algoritms. Machine Learning, 38, 3, 257-286.

YU, F. (1973). Introduction To Diffraction, information processing, and holography. England : The MIT Press.

Yu, F. ; Gregory, D. (1996). Optical Pattern Recognition: Architectures And Techniques. Proc. IEEE, Vol. 84, (1996), pp. 733.