EDITORES:

Manuel A. Serrano - Eduardo Fernández-Medina

Cristina Alcaraz - Noemí de Castro - Guillermo Calvo

Actas de las VI Jornadas Nacionales
(JNIC2021 LIVE)

INVESTIGACIÓN EN ciber segu ridad

Ediciones de la Universidad
de Castilla-La Mancha

Investigación en Ciberseguridad

Actas de las VI Jornadas Nacionales (JNIC2021 LIVE)

Online 9-10 de junio de 2021
Universidad de Castilla-La Mancha

Investigación en Ciberseguridad

Actas de las VI Jornadas Nacionales (JNIC2021 LIVE)

Online 9-10 de junio de 2021
Universidad de Castilla-La Mancha

Editores:
Manuel A. Serrano,
Eduardo Fernández-Medina,
Cristina Alcaraz
Noemí de Castro
Guillermo Calvo

Ediciones de la Universidad
de Castilla-La Mancha

Cuenca, 2021




Edita: Ediciones de la Universidad de Castilla-La Mancha

Colección JORNADAS Y CONGRESOS n.º 34

Esta editorial es miembro de la UNE, lo que garantiza la difusión y comercialización de sus publicaciones a nivel nacional e internacional.

I.S.B.N.: 978-84-9044-463-4
D.O.I.: http://doi.org/10.18239/jornadas_2021.34.00

Hecho en España (U.E.) – *Made in Spain (E.U.)*

Bienvenida del Comité Organizador

Tras la parada provocada por la pandemia en 2020, las VI Jornadas Nacionales de Investigación en Ciberseguridad (JNIC) vuelven el 9 y 10 de Junio del 2021 con energías renovadas, y por primera vez en su historia, en un formato 100% online. Esta edición de las JNIC es organizada por los grupos GSyA y Alarcos de la Universidad de Castilla-La Mancha en Ciudad Real, y con la activa colaboración del comité ejecutivo, de los presidentes de los distintos comités de programa y del Instituto Nacional de Ciberseguridad (INCIBE). Continúa de este modo la senda de consolidación de unas jornadas que se celebraron por primera vez en León en 2015 y le siguieron Granada, Madrid, San Sebastián y Cáceres, consecutivamente hasta 2019, y que, en condiciones normales se habrían celebrado en Ciudad Real en 2020.

Estas jornadas se han convertido en un foro de encuentro de los actores más relevantes en el ámbito de la ciberseguridad en España. En ellas, no sólo se presentan algunos de los trabajos científicos punteros en las diversas áreas de ciberseguridad, sino que se presta especial atención a la formación e innovación educativa en materia de ciberseguridad, y también a la conexión con la industria, a través de propuestas de transferencia de tecnología. Tanto es así que, este año se presentan en el Programa de Transferencia algunas modificaciones sobre su funcionamiento y desarrollo que han sido diseñadas con la intención de mejorarlo y hacerlo más valioso para toda la comunidad investigadora en ciberseguridad.

Además de lo anterior, en las JNIC estarán presentes excepcionales ponentes (Soledad Antelada, del Lawrence Berkeley National Laboratory, Ramsés Gallego, de Micro Focus y Mónica Mateos, del Mando Conjunto de Ciberdefensa) mediante tres charlas invitadas y se desarrollarán dos mesas redondas. Éstas contarán con la participación de las organizaciones más relevantes en el panorama industrial, social y de emprendimiento en relación con la ciberseguridad, analizando y debatiendo el papel que está tomando la ciberseguridad en distintos ámbitos relevantes.

En esta edición de JNIC se han establecido tres modalidades de contribuciones de investigación, los clásicos artículos largos de investigación original, los artículos cortos con investigación en un estado más preliminar, y resúmenes extendidos de publicaciones muy relevantes y de alto impacto en materia de ciberseguridad publicados entre los años 2019 y 2021. En el caso de contribuciones de formación e innovación educativa, y también de transferencias se han considerado solamente artículos largos. Se han recibido para su valoración un total de 86

contribuciones organizadas en 26, 27 y 33 artículos largos, cortos y resúmenes ya publicados, de los que los respectivos comités de programa han aceptado 21, 19 y 27, respectivamente. En total se ha contado con una ratio de aceptación del 77%. Estas cifras indican una participación en las jornadas que continúa creciendo, y una madurez del sector español de la ciberseguridad que ya cuenta con un volumen importante de publicaciones de alto impacto.

El formato online de esta edición de las jornadas nos ha motivado a organizar las jornadas de modo más compacto, distinguiendo por primera vez entre actividades plenarias (charlas invitadas, mesas redondas, sesión de formación e innovación educativa, sesión de transferencia de tecnología, junto a inauguración y clausura) y sesiones paralelas de presentación de artículos científicos. En concreto, se han organizado 10 sesiones de presentación de artículos científicos en dos líneas paralelas, sobre las siguientes temáticas: detección de intrusos y gestión de anomalías (I y II), ciberataques e inteligencia de amenazas, análisis forense y cibercrimen, ciberseguridad industrial, inteligencia artificial y ciberseguridad, gobierno y riesgo, tecnologías emergentes y entrenamiento, criptografía, y finalmente privacidad.

En esta edición de las jornadas se han organizado dos números especiales de revistas con elevado factor de impacto para que los artículos científicos mejor valorados por el comité de programa científico puedan enviar versiones extendidas de dichos artículos. Adicionalmente, se han otorgado premios al mejor artículo en cada una de las categorías. En el marco de las JNIC también hemos contado con la participación de la Red de Excelencia Nacional de Investigación en Ciberseguridad (RENIC), impulsando la ciberseguridad a través de la entrega de los premios al *Mejor Trabajo Fin de Máster en Ciberseguridad* y a la *Mejor Tesis Doctoral en Ciberseguridad*. También se ha querido acercar a los jóvenes talentos en ciberseguridad a las JNIC, a través de un CTF (Capture The Flag) organizado por la Universidad de Extremadura y patrocinado por Viewnext.

Desde el equipo que hemos organizado las JNIC2021 queremos agradecer a todas aquellas personas y entidades que han hecho posible su celebración, comenzando por los autores de los distintos trabajos enviados y los asistentes a las jornadas, los tres ponentes invitados, las personas y organizaciones que han participado en las dos mesas redondas, los integrantes de los distintos comités de programa por sus interesantes comentarios en los procesos de revisión y por su colaboración durante las fases de discusión y debate interno, los presidentes de las sesiones, la Universidad de Extremadura por organizar el CTF y la empresa Viewnext por patrocinarlo, los técnicos del área TIC de la UCLM por el apoyo con la plataforma de comunicación, los voluntarios de la UCLM y al resto de organizaciones y entidades patrocinadoras, entre las que se encuentra la Escuela Superior de Informática, el Departamento de Tecnologías y Sistemas de Información y el Instituto de Tecnologías y Sistemas de Información, todos ellos de la Universidad de Castilla-La Mancha, la red RENIC, las cátedras (Telefónica e Indra) y aulas (Avanttic y Alpinia) de la Escuela Superior de Informática, la empresa Cojali, y muy especialmente por su apoyo y contribución al propio INCIBE.

Manuel A. Serrano, Eduardo Fernández-Medina
Presidentes del Comité Organizador
Cristina Alcaraz
Presidenta del Comité de Programa Científico
Noemí de Castro
Presidenta del Comité de Programa de Formación e Innovación Educativa
Guillermo Calvo Flores
Presidente del Comité de Transferencia Tecnológica

Índice General

Comité Ejecutivo

Juan Díez González	INCIBE
Luis Javier García Villalba	Universidad de Complutense de Madrid
Eduardo Fernández-Medina Patón	Universidad de Castilla-La Mancha
Guillermo Suárez-Tangil	IMDEA Networks Institute
Andrés Caro Lindo	Universidad de Extremadura
Pedro García Teodoro	Universidad de Granada. Representante de red RENIC
Noemí de Castro García	Universidad de León
Rafael María Estepa Alonso	Universidad de Sevilla
Pedro Peris López	Universidad Carlos III de Madrid

Comité Organizador

Presidentes del Comité Organizador

Eduardo Fernández-Medina Patón	Universidad de Castilla-la Mancha
Manuel Ángel Serrano Martín	Universidad de Castilla-la Mancha

Finanzas

David García Rosado	Universidad de Castilla-la Mancha
Luis Enrique Sánchez Crespo	Universidad de Castilla-la Mancha

Actas

Antonio Santos-Olmo Parra	Universidad de Castilla-la Mancha

Difusión

Julio Moreno García-Nieto	Universidad de Castilla-la Mancha
José Antonio Cruz Lemus	Universidad de Castilla-la Mancha
María A Moraga de la Rubia	Universidad de Castilla-la Mancha

Webmaster

Aurelio José Horneros Cano	Universidad de Castilla-la Mancha

Logística y Organización

Ignacio García-Rodriguez de Guzmán	Universidad de Castilla-la Mancha
Ismael Caballero Muñoz-Reja	Universidad de Castilla-la Mancha
Gregoria Romero Grande	Universidad de Castilla-la Mancha
Natalia Sanchez Pinilla	Universidad de Castilla-la Mancha

Comité de Programa Científico

Presidenta

Cristina Alcaraz Tello	Universidad de Málaga

Miembros

Aitana Alonso Nogueira	INCIBE
Marcos Arjona Fernández	ElevenPaths
Ana Ayerbe Fernández-Cuesta	Tecnalia
Marta Beltrán Pardo	Universidad Rey Juan Carlos
Carlos Blanco Bueno	Universidad de Cantabria
Jorge Blasco Alís	Royal Holloway, University of London
Pino Caballero-Gil	Universidad de La Laguna
Andrés Caro Lindo	Universidad de Extremadura
Jordi Castellà Roca	Universitat Rovira i Virgili
José M. de Fuentes García-Romero de Tejada	Universidad Carlos III de Madrid
Jesús Esteban Díaz Verdejo	Universidad de Granada
Josep Lluis Ferrer Gomila	Universitat de les Illes Balears
Dario Fiore	IMDEA Software Institute
David García Rosado	Universidad de Castilla-La Mancha
Pedro García Teodoro	Universidad de Granada
Luis Javier García Villalba	Universidad Complutense de Madrid
Iñaki Garitano Garitano	Mondragon Unibertsitatea
Félix Gómez Mármol	Universidad de Murcia
Lorena González Manzano	Universidad Carlos III de Madrid
María Isabel González Vasco	Universidad Rey Juan Carlos I
Julio César Hernández Castro	University of Kent
Luis Hernández Encinas	CSIC
Jorge López Hernández-Ardieta	Banco Santander
Javier López Muñoz	Universidad de Málaga
Rafael Martínez Gasca	Universidad de Sevilla
Gregorio Martínez Pérez	Universidad de Murcia

David Megías Jiménez	Universitat Oberta de Cataluña
Luis Panizo Alonso	Universidad de León
Fernando Pérez González	Universidad de Vigo
Aljosa Pasic	ATOS
Ricardo J. Rodríguez	Universidad de Zaragoza
Fernando Román Muñoz	Universidad Complutense de Madrid
Luis Enrique Sánchez Crespo	Universidad de Castilla-La Mancha
José Soler	Technical University of Denmark-DTU
Miguel Soriano Ibáñez	Universidad Politécnica de Cataluña
Victor A. Villagrá González	Universidad Politécnica de Madrid
Urko Zurutuza Ortega	Mondragon Unibertsitatea
Lilian Adkinson Orellana	Gradiant
Juan Hernández Serrano	Universitat Politécnica de Cataluña

Comité de Programa de Formación e Innovación Educativa

Presidenta

Noemí De Castro García	Universidad de León

Miembros

Adriana Suárez Corona	Universidad de León
Raquel Poy Castro	Universidad de León
José Carlos Sancho Núñez	Universidad de Extremadura
Isaac Agudo Ruiz	Universidad de Málaga
Ana Isabel González-Tablas Ferreres	Universidad Carlos III de Madrid
Xavier Larriva	Universidad Politécnica de Madrid
Ana Lucila Sandoval Orozco	Universidad Complutense de Madrid
Lorena González Manzano	Universidad Carlos III de Madrid
María Isabel González Vasco	Universidad Rey Juan Carlos
David García Rosado	Universidad de Castilla - La Mancha
Sara García Bécares	INCIBE

Comité de Transferencia Tecnológica

Presidente

Guillermo Calvo Flores	INCIBE

Miembros

José Luis González Sánchez	COMPUTAEX
Marcos Arjona Fernández	ElevenPaths
Victor Villagrá González	Universidad Politécnica de Madrid
Luis Enrique Sánchez Crespo	Universidad de Castilla – La Mancha

Sesión de Investigación B1: Inteligencia Artificial en ciberseguridad
http://doi.org/10.18239/jornadas_2021.34.41

Entrenamiento optimizado de redes neuronales para reconocimiento biométrico

Gonzalo García Miranda
Universidad de León
Campus de Vegazana s/n, 24071, León
ggarcm04@estudiantes.unileon.es

Alberto Calvo García
Universidad de León
Campus de Vegazana s/n, 24071, León
acalvg05@estudiantes.unileon.es

Claudia Álvarez Aparicio
ORCID 0000-0002-7465-8054
Grupo de Robótica
Universidad de León
Campus de Vegazana s/n, 24071, León
calvaa@unileon.es

Ángel Manuel Guerrero Higueras
ORCID 0000-0001-8277-0700
Grupo de Robótica
Universidad de León
Campus de Vegazana s/n, 24071, León
am.guerrero@unileon.es

Francisco Javier Rodríguez Lera
ORCID 0000-0002-8400-7079
Grupo de Robótica
Universidad de León
Campus de Vegazana s/n, 24071, León
calvaa@unileon.es

Camino Fernández Llamas
ORCID 0000-0002-8705-4786
Grupo de Robótica
Universidad de León
Campus de Vegazana s/n, 24071, León
camino.fernandez@unileon.es

***Resumen*—Los sistemas actuales de autenticación biométrica para un grupo de individuos suelen requerir de un reentrenamiento completo cuando se registran nuevos usuarios, causando la indisponibilidad del sistema durante todo ese tiempo. En el intento de solventar dicho problema se propone un sistema de autenticación que permite autenticar a un usuario registrado con su huella dactilar sin afectar a la disponibilidad. Este sistema es gestionado por un algoritmo que estima y gestiona un cierto número necesario de redes neuronales que funcionan de forma paralela de modo que el proceso de alta de nuevos usuarios, además de ser rápido, no afecta en ningún caso al funcionamiento del sistema ni al resto de usuarios.**

***Palabras Clave*—Autenticación, biometría, huella dactilar, redes neuronales.**

Tipo de contribución: *Investigación original.*

I. Introducción

En la actualidad son frecuentes los dispositivos que utilizan un sistema de autenticación biométrica para identificar a sus usuarios y proteger así la información de éstos. La popularización y normalización del reconocimiento biométrico ha contribuido a la aceptación general de este tipo de autenticación por parte de los usuarios, siendo cada vez más común en organizaciones y empresas.

Los sistemas biométricos aplicados en la autenticación de un grupo de usuarios habitualmente usan técnicas de aprendizaje automático y/o redes neuronales para identificar a los sujetos. En estos casos, los procesos de alta de nuevos usuarios suelen ser bastante costosos, tanto en tiempo como en operatividad, puesto que requieren de un nuevo entrenamiento para incluir a los últimos individuos registrados, haciendo que el sistema quede indisponible temporalmente para los demás o, en su defecto, para los nuevos (ya que todavía no podrían hacer uso de él). Así pues, cuando la cantidad de usuarios aumenta el problema se va agravando con tiempos de indisponibilidad cada vez más elevados, pudiendo incluso consumir demasiados recursos. La solución más común consiste en posponer el alta de los nuevos usuarios a determinados momentos en los que esta indisponibilidad del sistema no afecte al funcionamiento general de la red. De esta manera, aunque es cierto que el funcionamiento del sistema para el resto de usuarios no se ve comprometido en ningún caso, se alarga el proceso de alta más de lo estrictamente necesario en tiempos de computación.

La idea aquí presentada consiste en compartimentar el sistema de autenticación biométrica de los usuarios en una serie de subsistemas, dedicados cada uno a un subconjunto tomado de entre el total de los usuarios registrados en el sistema. De este modo se pretende garantizar la disponibilidad del sistema para todos los usuarios, reduciendo directamente los tiempos de entrenamiento mediante un reentrenamiento parcial que incluya al nuevo individuo casi inmediatamente. En consecuencia, se minimiza el tiempo de espera en el que el nuevo usuario registrado puede hacer un uso normal y continuado del sistema, sin afectar a la disponibilidad para el resto de usuarios. Además, análogamente, este procedimiento permite dar de baja a un usuario prácticamente de forma instantánea.

Este trabajo tiene como objetivo principal el diseño de un algoritmo, denominado Algoritmo Gestor de Redes Neuronales (AG-RN), que permite gestionar las redes neuronales que conforman el sistema de autenticación biométrico para un grupo de individuos, de manera que se puedan dar de alta nuevos (o de baja antiguos) usuarios de forma casi inmediata garantizando la disponibilidad constante del sistema. En concreto, se ha desarrollado una prueba de concepto (PoC, por sus siglas en inglés) para reconocimiento de huella dactilar con el fin de demostrar el correcto funcionamiento, la efectividad general y la reducción de los tiempos de entrenamiento ante el aumento del número de individuos registrados.

La estructura de este artículo es la siguiente. En la sección II se expone una breve revisión metódica de otras posibles soluciones a situaciones similares para poner en contexto el problema en cuestión y la solución planteada. En la sección III se explica el procedimiento desarrollado en relación al conjunto de datos utilizado y al diseño de la arquitectura del

sistema a implementar. En la sección IV se presenta el análisis de la solución aplicada, detallando los criterios en cuanto a tiempos de entrenamiento y la precisión del sistema requeridos para luego diseñar el algoritmo que finalmente se implementa en la PoC. En la sección V se describen las pruebas de verificación para demostrar que la PoC es correcta y, por tanto, satisfactoria. En la sección VI se infiere la importancia de la solución planteada junto con sus posibles inconvenientes y líneas de investigación futuras.

II. Estado del arte

Fundamentalmente los procedimientos de autenticación se realizan mediante información secreta (como contraseñas), objetos especiales (como tarjetas personales), o bien atributos biométricos (como huellas dactilares). Cada una de estas formas de autenticación conlleva diferentes grados de seguridad, puesto que la información secreta puede ser olvidada, robada o compartida; y los objetos especiales pueden ser sustraídos, deteriorados o duplicados. Sin embargo, los atributos biométricos de una persona son únicos, difícilmente usurpables o replicables y, salvo accidente grave, inmutables. No obstante, a medida que aumenta la seguridad que proporcionan estos métodos también se incrementa la complicación en su implementación. En particular, la captación de los atributos biométricos requiere de un dispositivo complejo capaz de analizar el cuerpo, realizando así una lectura de datos que posteriormente son tratados y codificados en el formato deseado, la llamada plantilla biométrica [1].

La principal desventaja de los sistemas de autenticación biométrica son las interferencias en la lectura de datos, pudiendo arrojar una mayor tasa de falsos negativos que el resto de tipos de autenticación. Sin embargo, se pueden adaptar las plantillas de identificación con los sucesivos usos lícitos del sistema como solución [2]. Por otro lado, aunque en general los datos biométricos son inalterables, hay trabajos que intentan conseguir la identificación mediante diversas modificaciones [3], [4]. Además, esta inmutabilidad origina una contingencia inherente conocida como riesgo de irrevocabilidad, que se refiere a la incapacidad de actualizar, reeditar o destruir el atributo biométrico una vez éste haya sido comprometido. Para mitigar este riesgo una opción común es la autenticación multifactor [5], generando plantillas biométricas dependientes de otros factores como, por ejemplo, una contraseña y que, por tanto, sí permiten la actualización, reedición o destrucción de éstas [6]. En particular, también es importante destacar que los rasgos biométricos se consideran datos de carácter personal a todos los efectos legales. En consecuencia su uso, almacenamiento o tratamiento está sometido al cumplimiento de las distintas exigencias de carácter jurídico, técnico, físico y organizativo como, por ejemplo, prevee el Real Decreto de la Ley Orgánica de Protección de Datos de Carácter Personal (RDLOPD) [7].

La clasificación de los sistemas de identificación biométrica está basada, sustancialmente, en: atributos físicos (como huella dactilar o palmar, reconocimiento facial, escáner de iris o retina, etc.) o comportamiento (como el reconocimiento de voz, firma o tecleo). Principalmente, el más utilizado es el atributo físico de huella dactilar, quizás por su carácter orgánico único, ya que supone un desarrollo diferente para cada individuo incluso entre los propios dedos.

Tal y como se describe en los manuales originarios del tratamiento de huellas dactilares, los primeros trabajos se centraron en la detección y comparación de las características propias de cada huella y, más particularmente, en los diferentes patrones que forman las pequeñas rugosidades de las palmas de las manos y los dedos [8]. De este modo, mediante el reconocimiento de patrones en la huella, la identificación es inequívoca asignando a cada entrada una puntuación basada en las características genuinas de la huella que pertenecen a un solo individuo [9]. Un método común de reconocimiento de patrones es la identificación de las propias *minutiaes*, que son las líneas que según su grosor, separación y formas configuran estos patrones. Su distancia y sus intersecciones o falta de ellas permite definir de un modo más fino estos puntos de comparación entre huellas, lo cual suele inducir resultados bastante satisfactorios [10]. Ahora bien, este método difiere un poco del reconocimiendo de patrones en sí mismo siendo más próximo a lo que hace una red neuronal convolucional (CNN, del inglés *Convolutional Neural Network*). El motivo principal se basa en la necesidad de preprocesar la imagen de la huella antes de su tratamiento, para encontrar así esos puntos asignándoles un cierto valor que luego es comparado con los obtenidos de la huella a autenticar. Existen muchos métodos diferentes, continuándose avanzando actualmente en los métodos de detección, mejora de imágenes y codificación de las mismas [11], [12], [13]. Además, la manera de obtener huellas dactilares puede ser muy variada influyendo en la calidad de la imagen recopilada, pudiendo afectar gravemente a los procedimientos de identificación y codificación de *minutiaes* (ya que los errores de lectura pueden ser interpretados como rasgos propios). Aunque ambos métodos son ampliamente utilizados y obtienen buenos resultados, algunos trabajos afirman que el reconocimiento de patrones puede ser más correcto al abstraerse de ese proceso de extracción de *minutiaes*, puesto que requiere de una codificación adicional [14].

Las CNNs se pueden definir como la replicación artificial de la estructura cerebral que posibilita el aprendizaje mediante un conjunto estratificado de capas de nodos que filtran sucesivamente información para el análisis de datos visuales, destacando su aplicación en visión por ordenador como en la clasificación de imágenes y el reconocimiento de objetos. Así pues, este tipo de red neuronal se puede emplear en la identificación de huellas dactilares mediante las *minutiaes*. Este método utiliza toda la imagen de la huella aplicando sucesivas capas de convolución que, a modo de filtros, permiten distinguir sucesivamente patrones más específicos dentro de la imagen. Su uso requiere de un procesamiento previo de la imagen para normalizarla, limpiar el posible ruido y estilizar las líneas que forman estas *minutiaes* o definir una serie de patrones a identificar dentro de la huella, sin tener así que declararlo después de forma explícita [15]. De esta manera se pueden identificar y extraer los puntos de la imagen para luego calcular sus distancias relativas y orientaciones, siendo finalmente los parámetros de entrada de la red neuronal [12], [16], [17]. Así pues, se pueden crear *autoencoders* mediante la sucesiva aplicación de capas de convolución en conjunción con capas de "Max Pooling". Estos *autoencoders* crean unos descriptores de las imágenes mediante un aprendizaje no supervisado que pueden utilizarse para, entre otras cosas,

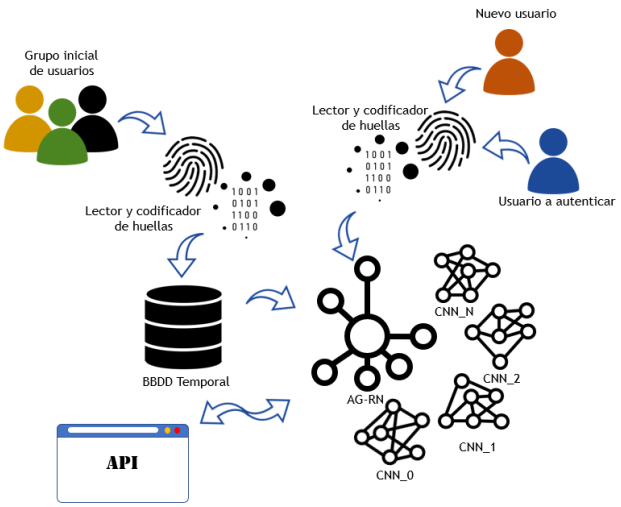

Figura 1. Esquema ideal del sistema.

tomar automáticamente las características más importantes de una imagen y, a partir de ellas, relacionar dos imágenes tomadas desde ángulos distintos o con alguna modificación distinta entre ambas [18]. Todas estas características que aportan las CNN y su uso para crear *autoencoders* permiten un reconocimiento de patrones sin la necesidad de explicitarlos, lo cual puede hacerse extensivo al reconocimiento de patrones dentro de las propias huellas dactilares. Es decir, tal y como demuestran Chowdhury *et al.* [15], es posible entrenar una CNN para que realice reconocimiento de huellas dactilares mediante los patrones que las *minutiaes* forman en ellas, sin que la propia red sea "consciente" de que eso es lo que está haciendo.

III. Metodología

Como el objetivo es implementar un sistema de autenticación mediante huella digital que permite identificar individuos realizando el registro o borrado de usuarios en el momento (es decir, sin requerir de un reentrenamiento total), es preciso, por un lado, recolectar un conjunto de datos de huellas dactilares lo suficientemente grande y, por otro lado, diseñar una arquitectura conveniente de las CNNs para que el algoritmo AG-RN pueda gestionar el número variable de redes con una interfaz que permita la utilización de este sistema por parte de terceros. En la Fig. 1 se muestra el esquema ideal del sistema.

El conjunto de datos utilizado proviene del recolectado en una base de datos preexistente, debido a que no se ha dispuesto de un procedimiento para ello. Además, las CNNs necesitan tener un tamaño manejable (en términos computacionales) y cada una debe ser entrenada con un subconjunto diferente y reducido de los datos totales de los individuos, pero manteniendo un nivel suficiente de redundancia de los datos de cada individuo a lo largo de varias de estas redes. En particular, el desarrollo general del sistema se ha realizado casi exclusivamente en el lenguaje Python por su uso común y flexible. Adicionalmente la interfaz del sistema es muy básica, a modo ilustrativo, consistente en el uso del intérprete Python de Jupyter para probar y evaluar de forma sencilla el algoritmo.

III-A. Conjunto de datos

La base de datos de huellas dactilares seleccionada para el sistema es Sokoto Coventry Fingerprint (SOCOFing) [19], que provee imágenes de huellas dactilares de cada uno de los 10 dedos de 600 personas. En especial, para mantener un conjunto grande de muestras, se ha decidido interpretar cada una de estas huellas como un individuo diferente, ya que a efectos de nuestro estudio en términos de autenticación se trata de diferentes plantillas biométricas. Además, SOCOFing provee de unas imágenes alteradas artificialmente de cada una de las huellas originales, simulando en tres grados distintos (fácil, medio y difícil) los tres tipos diferentes de alteraciones: obliteración, rotación central y corte en Z [3], [4]. No obstante, no es el objetivo de este trabajo centrarse en la identificación de sujetos cuyas huellas hayan sido alteradas, por lo que su uso principal aquí es para simular malas condiciones en el proceso de lectura que generan imágenes modificadas.

Las imágenes se proporcionan en formato BMP y sus dimensiones son 96x103 píxeles. Sin embargo, el marco de las imágenes ha sido recortado para dejar únicamente la imagen de la huella. Por tanto, la dimensión final es 90x90 píxeles leídas en escala de grises y almacenadas en ficheros NPZ, diferenciando cada uno de los conjuntos de datos proporcionados por SOCOFing (original, fácil, medio y difícil). Además, por cuestiones de eficiencia, se ha traducido el nombre de las imágenes a formato numérico donde el patrón general "ID_SEXO_MANO_DEDO_finger" para el siguiente ejemplo "11_M_Right_thumb_finger" se convierte en la etiqueta "[11, 0, 1, 0]". Estas etiquetas han sido almacenadas en archivos NPY, especiales para almacenar series de números NumPy.

Para comprobar diferentes funcionamientos de la red a lo largo de todo el proceso de desarrollo, se han creado tres subconjuntos de datos de diferentes tamaños del conjunto total de huellas que provee SOCOFing:

- El conjunto total de los datos (600 personas): Utilizado para el desarrollo de las CNNs que usa como base el AG-RN y también para el análisis de los tiempos de entrenamiento.
- Un conjunto con casi todos los datos (590 personas): Utilizado como conjunto de datos total sobre el que se crea el gestor y sobre el que se realizan los entrenamientos iniciales de las redes.
- Un conjunto con el resto de los datos (10 personas): Utilizado para añadir nuevos usuarios al gestor.

Además, se ha decidido utilizar la técnica de incremento de datos para aumentar el número de imágenes de huellas de cada individuo y garantizar así un buen entrenamiento. Esta técnica consiste en obtener, a partir de los datos de los que se dispone y mediante modificaciones artificiales de los mismos, un conjunto de datos mayor que permite un entrenamiento más extenso. Para ello se ha diseñado un proceso muy simple que ayuda a obtener una imagen algo distorsionada a partir de otra, lo cual permite simular múltiples lecturas de una misma huella por parte de un sensor con las pequeñas imperfecciones y/o modificaciones que la propia lectura pueda introducir. Se ha hecho uso de la librería imgaug [20], en concreto de

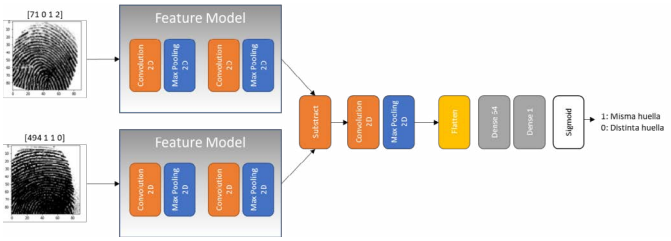

Figura 2. Arquitectura de la CNN.

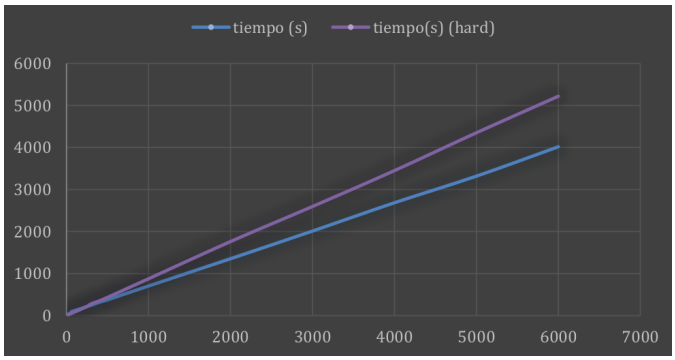

Figura 3. Tiempos de entrenamiento por volumen de usuarios.

augmenters, para asegurar que la imagen obtenida no se va a repetir mediante las siguientes técnicas:

- Escalar: Modifica el tamaño entre 0.9 y 1.1 veces el original.
- Traslación: Modifica la posición para que los mismos puntos no se encuentren en los mismos píxeles.
- Rotación: Se gira entre $-30°$ y $30°$.
- Desenfoque Gaussiano: Combina en cada pixel aquellos que lo rodean.

III-B. Arquitectura de la CNN

Una vez leído y codificado el conjunto de datos, se construye la red neuronal basada en reconocimiento de patrones de las *minutiaes* mediante convolución sin tener así que especificarlo explícitamente. Esta construcción se inspira en la propuesta por Chowdhury *et al.* [15].

La CNN está diseñada para recibir 2 imágenes de 90x90 y devolver la probabilidad de que las imágenes coincidan con la misma huella. Sobre cada una de las entradas se extraen sus características por medio de una red de extracción de características creada mediante la aplicación sucesiva de dos capas combinadas de "Convolution 2D" y "Max Pooling 2D". De este modo se generan los *autoencoders* necesarios para crear los descriptores sobre las distintas imágenes de las huellas dactilares. En particular, se aplica dos veces sobre las imágenes de entrada para distinguir, primero, el fondo de la huella y, posteriormente, hacer una distinción más fina dentro de la huella que permita captar los patrones que forman las *minutiaes*. Así pues, estas redes de extracción de características son las que crean la entrada que se aplica a la red que finalmente da la predicción. Esta segunda red recibe ambas entradas codificadas, dado que necesita comparar los patrones de características de ambas imágenes mediante la capa Substract. A continuación, se aplica de nuevo una capa de "Convolution 2D" y "Max Pooling 2D" para realizar una extracción de características de la comparación. Este nuevo descriptor proporciona una salida en 2D, que se aplana mediante una capa Flatten para pasarla a una nueva capa densa de 64 neuronas. La salida de esta última capa es una única neurona que da el resultado final, la predicción entre 0 y 1 que estima la probabilidad de acierto. En la Fig. 2 se muestra esta arquitectura de la CNN.

Antes de utilizar el modelo es necesario compilarlo, para lo que se necesitan elegir ciertos parámetros que se utilizan durante el entrenamiento de la red. En este caso como función de pérdidas se ha escogido la binary_crossentropy, dado que se usa en los modelos de clasificación binarios como el que se utiliza aquí, y como optimizador se ha seleccionado Adam [21].

IV. Implementación

Una vez creada la red y el modelo se prueba su funcionamiento. Primero, como es común en el entrenamiento de redes neuronales, se divide el conjunto de datos en datos de entrenamiento y datos de prueba. En este caso se ha hecho una división 90 %-10 % respectivamente. Ahora bien, los datos almacenados son huellas con su correspondiente etiqueta. Dado que el objetivo de nuestra red es el de comparar dos huellas y mostrar el grado de certidumbre en un rango de 0 a 1, se debe hacer un cambio para poder dar estos datos de forma correcta. Se trata de, para cada una de las huellas que están en el conjunto de datos tomar dos huellas aumentadas, una creada a partir de la misma huella y otra tomada de forma aleatoria entre el resto de datos y asignar a cada par de huellas un 1 o un 0 dependiendo de si es la misma huella o no, lo cual se hace mediante una clase adicional.

IV-A. Estudio de tiempos de entrenamiento y precisión

Para comprobar en qué medida un aumento en el tamaño del conjunto de datos está influyendo tanto en los tiempos de entrenamiento como en la precisión de las redes que se entrenan, se ha diseñado un experimento haciendo lo siguiente:

1. Se crean conjuntos de datos de distintos tamaños que oscilan desde los 20 a los 600 individuos, tomados de manera aleatoria de entre la totalidad de los individuos que conforman los 600 individuos de SOCOFing.
2. Para cada uno de esos conjuntos de datos se entrena el modelo utilizando las mismas condiciones salvo el propio conjunto y se guardan los tiempos de entrenamiento.
3. Se entrenan los modelos incluyendo tanto las huellas con alteraciones difíciles como sin ellas, dado que esto afecta a los tiempos de entrenamiento al haber más lotes en cada período y, sobre todo, al ratio de falso rechazo (FRR, del inglés *False Rejection Rate*), puesto que algunas alteraciones son bastante extremas.
4. Se prueba el modelo con 100 resultados positivos tomados de entre los valores de testeo del modelo y con otros 100 resultados negativos, lo cual permite obtener el FRR y el ratio de autenticación errónea (FAR, del inglés *False Acceptance Rate*) de cada una de las redes.

La evolución de los tiempos de entrenamiento del experimento se muestra en la Fig. 3.

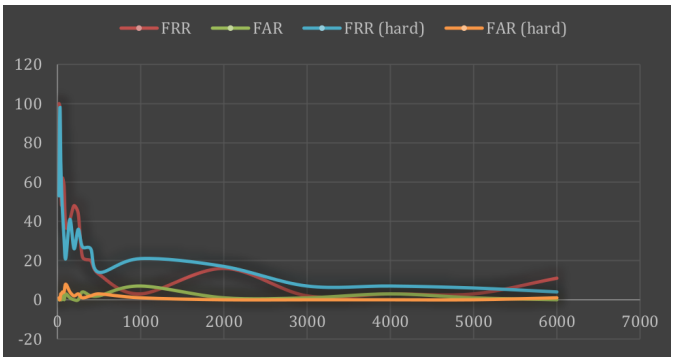

Figura 4. FAR y FRR por volumen de usuarios.

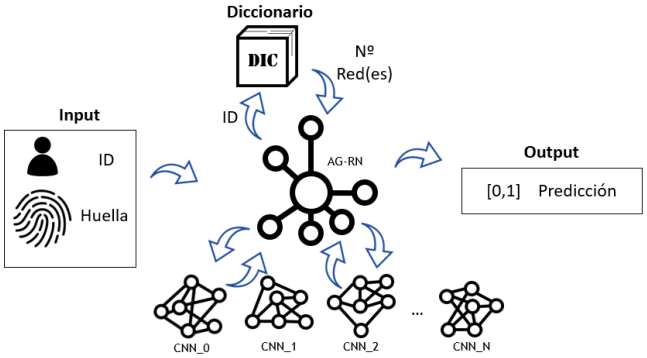

Figura 5. Diagrama del funcionamiento de la predicción del AG-RN.

Aunque este estudio de tiempos se ha realizado con un portátil estándar actual, estos resultados sirven para ilustrar la tendencia lineal general según aumenta el número de individuos. Asimismo, para las mismas redes que fueron entrenadas en el análisis de los tiempos de entrenamiento, se analiza la precisión de las predicciones cuando se produce este aumento en el número de individuos. Los resultados del FRR y del FAR se muestran en la Fig. 4.

Así pues, se observa que, aunque hay una cierta inestabilidad en redes con muy pocos individuos, la tendencia es claramente decreciente según se aumenta el número de usuarios incluidos en el entrenamiento. Las redes con muy pocos individuos presentan tasas de FRR muy elevadas, lo cual afectaría en gran medida a su usabilidad, mientras que para redes más grandes se mantiene en valores muy bajos, lo cual implica un funcionamiento muy certero de la red. En cuanto a la tasa de FAR se aprecia que, en general, se mantiene muy baja en todo momento salvo algún incremento puntual que se puede relacionar con una disparidad estadística. Por otro lado, tal y como se esperaba, el FAR es mayor en los conjuntos de datos que emplean las huellas con alteraciones difíciles, aunque con mayores conjuntos, las tasas tienden a igualarse. En definitiva, a partir de un número mínimo de usuarios la precisión de la red se mantiene en valores satisfactorios mientras que el tiempo de entrenamiento aumenta progresivamente según lo haga el número de individuos.

IV-B. Diseño del AG-RN

El primer paso para desarrollar el algoritmo es encontrar el Punto de Distribución (PD), entendido como el número óptimo de usuarios con el que la red tiene una respuesta correcta manteniendo un tiempo de entrenamiento aceptable. Entonces, el PD es variable para cada implementación del algoritmo en función de qué parámetro se considere más importante (es decir, un FAR bajo implica un PD alto mientras que un entrenamiento rápido implica un PD más bajo). Por tanto, hay que garantizar un balance para que la precisión de la red se mantenga como mínimo en un nivel aceptable antes de que el conjunto de datos sea demasiado grande. Por ejemplo, en el estudio realizado se ha decidio establecer un tiempo de entrenamiento máximo por red de 600 segundos y un FRR máximo del 30 %. Tras realizar un análisis sobre cómo afecta el tamaño de las redes a la precisión y el tiempo de entrenamiento, se ha obtenido un intervalo para el PD de [200, 700] y, seleccionando el valor más óptimo para el propósito de la prueba, el PD establecido es 250. Así pues, se subdivide el conjunto inicial en grupos de N usuarios, con N igual o muy cercano al PD seleccionado. No obstante, se precisa distribuir a cada usuario en al menos un subconjunto con el que se haya entrenado una red, de modo que así se puede autenticar al usuario de forma tan eficaz como con una red entrenada con el total de usuarios. En consecuencia, como requisito adicional, se hace necesario saber a qué subconjunto pertenece cada usuario mediante el uso, en este caso, de un diccionario.

El diccionario devuelve la(s) red(es) en la(s) que se encuentra el usuario. Esto representa la redundancia (Rd), ya que indica el número mínimo de redes en las que se debe incluir cada usuario, siendo también su valor ideal variable dependiendo de la implementación del AG-RN. La redundancia presenta varias funciones dentro del algoritmo, permitiendo la adicción inmediata de nuevos usuarios únicamente añadiendo y entrenando Rd nuevas redes. En este caso, sólo se debe controlar que tras sucesivas adicciones el número de redes no sea excesivo, puesto que esto afectaría al tiempo de respuesta del sistema. De la misma forma, también permite la eliminación de usuarios mediante la supresión de las redes en las que se encuentren y reentrenando nuevas redes para aquellos usuarios cuya redundancia haya caído por debajo de la establecida. Además, al tener no una sino Rd redes haciendo una predicción sobre los mismos datos de entrada, pero entrenadas con diferentes conjuntos de datos, las FFR y FAR son reducidas. Adicionalmente, una redundancia eficiente proporciona flexibilidad en el entrenamiento (ya que en los sucesivos entrenamientos de las redes por la adicción o eliminación de usuarios la mejora es sustancial, aunque es posible que en el inicial resulte mayor que entrenando una única red) y en el uso de memoria (debido a que se pueden cargar en primer lugar sólo aquellas redes en las que se encuentren los individuos que hagan un uso intensivo del sistema).

En la Fig. 5 se muestra un diagrama del funcionamiento del AG-RN.

Entonces, para distribuir de forma óptima los usuarios con una Rd determinada entre el total de redes de tamaño PD, se hace necesario un algoritmo que se ha denominado distribución sacudida. Este procedimiento consiste en, primero, distribuir los usuarios de forma secuencial en el número calculado de subconjuntos necesarios para garantizar la Rd

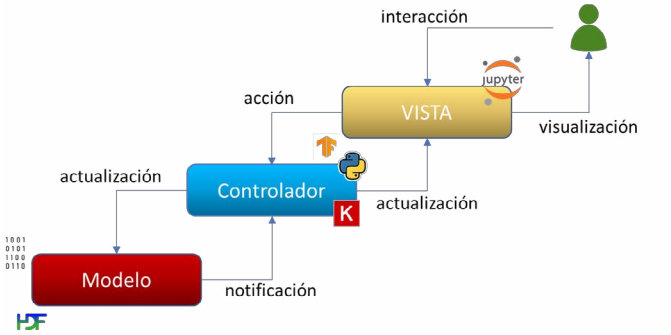

Figura 6. MVC aplicado en la PoC.

establecida (sin que éstos se repitan en el mismo subconjunto); y, segundo, realizar un intercambio semialeatorio entre los distintos subconjuntos (sin que éstos acaben estando repetidos en el mismo subconjunto) tantas veces como establece la Ec. 1, evitando así que muchos usuarios compartan todos sus subconjuntos lo que invalidaría la mayor parte de los beneficios de la redundancia.

$$\max\{\text{N.º Redes, PD}\} \cdot \text{PD}^2. \tag{1}$$

En el proceso de adición de nuevos usuarios, análogamente, la distribución de las nuevas redes consiste en añadir al propio usuario y después completar el subconjunto hasta obtener los PD usuarios, eligiendo entre el resto de forma aleatoria y evitando la repetición dentro del conjunto.

IV-C. Implementación de la PoC

La PoC implementada se encuentra en el siguiente repositorio público y accesible mediante licencia Creative Commons Zero: https://github.com/GonGMiranda/AG-RN.

Esta PoC exige los siguientes requisitos para poder validar el funcionamiento del AG-RN:

- Requisitos funcionales: Permitir crear un gestor que actúe sobre un conjunto inicial de datos con parámetros PD y Rd para realizar el proceso de distribución sacudida, el entrenamiento de redes, y la adición y eliminación de usuarios; guardar su estado y recuperarlo para las redes ya entrenadas; y crear un predictor que actúe sobre el gestor.
- Requisitos no funcionales: Permitir la predicción en paralelo de distintas instancias del gestor sin que se produzcan interrupciones entre ellas.

En cuanto a la interacción, se hace uso de un patrón de diseño clásico como es el Modelo-Vista-Controlador (MVC). Este diseño permite separar la interacción con el usuario, del control y modificación de los datos, y de su persistencia. Se utiliza como modelo el propio sistema de ficheros de Windows, almacenando el propio gestor en ficheros binarios y las redes entrenadas en ficheros HDF5. En referencia a la vista no se ha desarrollado una interfaz puesto que se hace uso de los cuadernos de Jupyter que permiten actuar de forma sencilla y visual para mostrar los resultados. El controlador está programado en Python, haciendo uso de la librería Keras, por su alto nivel de abstracción, y de TensorFlow, por ser la opción por defecto. En la Fig. 6 se puede observar el MVC aplicado en esta PoC.

Además, se crea una clase Predictor con un método sobre la que se ha aplicado el patrón Monostate [22], para predecir huellas dactilares. De esta manera los predictores se crean de forma paralela con el único propósito de hacer consultas sobre instancias distintas de la clase NNManager del gestor, sin posibilidad de modificar su estructura ni forzar la creación y borrado de usuarios o el entrenamiento de redes. Así pues, idealmente, el uso de un Predictor será único para cada huella que se quiera autenticar. Esto permitiría que cada uso contenga la red actualizada al momento de su utilización, aunque también puede realizarse la actualización periódicamente.

V. Evaluación

Con la intención de determinar que la implementación de la PoC ha sido realizada correctamente, se han diseñado una serie de casos de prueba como método de evaluación. La demostración de que los resultados de los casos de prueba han sido satisfactorios se encuentra en los cuadernos de Jupyter del repositorio de la implementación. Por tanto, a continuación sólo se mencionan tales casos de prueba, que en particular todos han sido verificados como satisfactorios lo que permite concluir que el software desarrollado del AG-RN en la PoC es correcto:

- Creación del conjunto de datos (fichero FPPreprocess.ipynb): La lectura de una imagen y su codificación es correcta, extrayéndose la etiqueta que identifica a su usuario. Además, la imagen codificada y su etiqueta se guardan y recuperan adecuadamente. Asimismo, se leen y codifican todas las imágenes y etiquetas tanto de una carpeta de huellas reales como de una de huellas alteradas, guardándose en los ficheros correspondientes.
- Desarrollo CNN (ficheros RNC.ipynb y Training stats.ipynb): El modelo de incremento de datos produce correctamente 9 imágenes distintas a partir de la original. El modelo de CNN se crea y entrena debidamente con un funcionamiento apropiado de la red tanto para una prueba positiva como para una negativa de una huella aleatoria. Además, se genera un conjunto de datos con huellas reales y alteradas de un número determinado de usuarios y se verifica que el FRR y el FAR son menores al 5 % y al 2 % respectivamente. Finalmente, se generan varios conjuntos de datos aleatorios y se entrenan distintas redes con ellos satisfactoriamente almacenando los resultados en un archivo Excel.
- Desarrollo AG-RN (fichero NNManager.ipynb): Se crea un nuevo NNManager configurado con los parámetros indicados y se carga adecuadamente, comprobándose que la distribución de los usuarios es correcta. Así pues, se verifica la autenticación de un usuario obteniéndose la predicción cuando las redes han sido parcialmente entrenadas y el rechazo en el otro caso. Además, se entrenan todas las redes del gestor y se guardan en sus respectivos ficheros HDF5 para verificar la autenticación negativa tanto para un usuario con una huella errónea como para otro cuya huella no pertenece al gestor correspondiente. Por último, se añade correctamente a un nuevo usuario, pudiéndose autenticar después de forma adecuada, y también se borra a un usuario satisfactoriamente.
- Predictor (fichero PredictorAG-RN.ipynb): Se verifica que el predictor se crea sobre la red entrenada y que

autentica al usuario convenientemente. Además, se comprueba que no se autentica a un usuario inválido, y que cuando el predictor no se crea se alerta mediante el error de que el gestor no existe.

VI. Conclusiones

La autenticación de usuarios es, actualmente, uno de los procesos más importantes en la seguridad de la información. Por tanto, cada vez es más común el uso de métodos más difíciles de vulnerar como el uso de factores biométricos. No obstante, a veces la inmediatez en su aplicación se encuentra lejos de lo deseable, como por ejemplo al autenticar a un grupo de usuarios en una organización mediante el uso de huellas dactilares.

En este trabajo se presenta AG-RN, un algoritmo que permite la adición o eliminación de usuarios a un sistema de reconocimiento biométrico basado en redes neuronales sin ser necesario un reentrenamiento de la totalidad de la red. La potencia de este algoritmo está basada en la capacidad de gestión de varias redes neuronales, ofreciendo un balance óptimo entre el tamaño de cada una de estas redes que están entrenadas con subconjuntos diferentes del total de individuos, y la redundancia de los datos de cada individuo distribuido entre varias de estas redes. Así pues, se ha comprobado que se puede crear una red neuronal basada en capas de convolución y "Max Pooling" capaz de identificar si dos huellas dactilares son o no del mismo dedo de un mismo individuo, incluso aunque la lectura de las mismas no sea perfecta o que la huella haya sufrido alteraciones que conlleven una modificación visual en el patrón general formado por las *minutiaes*. Asimismo, se ha mostrado que el rendimiento de esta red empeora al ser entrenada con conjuntos más reducidos de individuos y que el tiempo de entrenamiento aumenta de forma lineal al aumentar el conjunto de usuarios.

Además, para demostrar la viabilidad de este algoritmo, se ha desarrollado una PoC en Python que hace uso de un modelo basado en una CNN que permite la autenticación de usuarios mediante su huella dactilar. En particular, esta PoC utiliza una base de datos preexistente de un conjunto de imágenes de huellas dactilares, y no incluye una interfaz de usuario sino que las pruebas pertinentes se han realizado en cuadernos de Jupyter, ya que son una forma más sencilla pero también visual de mostrar los resultados.

El repositorio del sistema de autenticación presentado es público y tiene una licencia libre con la intención de servir como material de apoyo a otros investigadores y, también, que pueda ser completada en el futuro con otras nuevas funcionalidades y diversas mejoras como, por ejemplo, un proceso de distribución de redes más eficiente, la inclusión de otros métodos de autenticación o la extensión del algoritmo a opciones multifactor.

Finalmente, se considera que el algoritmo propuesto puede dar solución a un problema real, demostrando que un sistema de autenticación biométrica basado en una implantación del AG-RN permite registrar o eliminar a un usuario prácticamente de forma inmediata, sin afectar al funcionamiento general del sistema o a los usuarios creados anteriormente.

Referencias

[1] A. K. Jain, A. Ross y S. Prabhakar: "An Introduction to Biometric Recognition", en *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4-20, 2004. DOI: 10.1109/TCSVT.2003.818349.

[2] R. Noval, y F. Perales: "Adaptative Templates in Biometric Authenticacion", en *WSCG International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2014.

[3] J. Feng, A. K. Jain y A. Ross: "Detecting Altered Fingerprints", en *2010 20th International Conference on Pattern Recognition*, pp. 1622-1625, 2010. DOI: 10.1109/ICPR.2010.401.

[4] E. Tabassi, T. Chugh, D. Deb, y A. K. Jain: "Altered Fingerprints: Detection and Localization", en *repositorio arXiv*, Cornell University, 2018. ArXiv: 1805.00911v2 [cs.CV].

[5] J. R. Hamlet y L. G. Pierson: "Multi-Factor Authentication", *Sandia Corporation*, Patente US8868923B1, Estados Unidos, 2010.

[6] S. H. Khan, M. A. Akbar, F. Shahzad, M. Farooq y Z. Khan: "Secure Biometric Template Generation for Multi-Factor Authentication", en *Pattern Recognition*, vol. 48, n. 2, pp. 458-472, 2015. DOI: 10.1016/j.patcog.2014.08.024.

[7] España, Ministerio de Justicia: "Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal", en *Boletín Oficial del Estado*, 19 de enero de 2008, n. 17, pp. 4103-4136. Referencia: BOE-A-2008-979.

[8] F. Galton: "Finger Prints", Londres: *McMillan*, 1892.

[9] H. Abdulkareem: "Fingerprint Identification System using Neural Networks", en *College of Engineering Journal (NUCEJ)*, Nahrain University, vol. 15, pp. 234-244, 2012.

[10] F. Pernus, S. Kovacic y L. Gyergyek: "Minutiae-Based Fingerprint Recognition", en *Proceedings of the Fifth International Conference on Pattern Recognition*, pp. 1380-1382, 1980.

[11] D. Maio y D. Maltoni: "Direct Gray-Scale Minutiae Detection in Fingerprints", en *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 27-40, 1997. DOI: 10.1109/34.566808.

[12] N. Zaeri: "Minutiae-Based Fingerprint Extraction and Recognition", en *Biometrics*, 2011. DOI: 10.5772/17527.

[13] A. Farina, Z. M. Kovács-Vajna y A. Leone: "Fingerprint Minutiae Extraction from Skeletonized Binary Images", en *Pattern Recognition*, vol. 32, n. 5, pp. 877-889, 1999. DOI: 10.1016/S0031-3203(98)00107-1.

[14] S. Narwal y D. Kaur: "Comparison between Minutiae Based and Pattern Based Algorithm of Fingerprint Image", en *International Journal of Information Engineering and Electronic Business*, vol. 8, n. 2, pp. 23-29, 2016. DOI: 10.5815/ijieeb.2016.02.03.

[15] A. Chowdhury, S. Kirchgasser, A. Uhl y A. Ross: "Can a CNN Automatically Learn the Significance of Minutiae Points for Fingerprint Matching?", en *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 340-348, 2020. DOI: 10.1109/WACV45572.2020.9093301.

[16] T. R. Borah, K. K. Sarma y P. H. Talukdar: "Fingerprint Recognition using Artificial Neural Network", en *International Journal of Electronics Signals and Systems (IJESS)*, vol. 3, n. 1, pp. 98-101, 2013.

[17] F. A. A. Minhas, M. Arif y M. Hussain: "Fingerprint Identification and Verification System using Minutiae Matching", en *National Conference on Emerging Technologies*, pp. 141-147, 2004.

[18] L. Chen, F. Rottensteiner y C. Heipke: "Feature Descriptor by Convolution and Pooling Autoencoders", en *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3/W2, pp. 31-38, 2015. DOI: 10.5194/isprsarchives-XL-3-W2-31-2015.

[19] Y. I. Shehu, A. Ruiz-García, V. Palade y A. James: "Sokoto Coventry Fingerprint Dataset (SOCOFing)", en *repositorio arXiv*, Cornell University, 2018. ArXiv: 1807.10609v1 [cs.CV].

[20] A. Jung: "Imgaug", en *repositorio GitHub*, 2020. URL: https://github.com/aleju/imgaug.

[21] D. P. Kingma y J. L. Ba: "Adam: A Method for Stochastic Optimization", en *3rd International Conference for Learning Representations (ICLR)*, 2015. ArXiv: 1412.6980 [cs.LG].

[22] Wiki: "Monostate Pattern" [recurso web]. URL: https://wiki.c2.com/?MonostatePattern.