

UN ACERCAMIENTO A LA INDUSTRIA 4.0 A TRAVÉS DE REDES NEURONALES PARA LA REDUCCIÓN DE SCRAP EN UNA EMPRESA AUTOMOTRIZ

AN APPROACH TO INDUSTRY 4.0 THROUGH NEURAL NETWORKS FOR SCRAP REDUCTION IN AN AUTOMOTIVE COMPANY

Norma Verónica Ramírez Pérez

Tecnológico Nacional de México en Celaya, México
norma.ramirez@itcelaya.edu.mx

Martín Laguna Estrada

Tecnológico Nacional de México en Celaya, México
martin.laguna@itcelaya.edu.mx

Norma Natalia Rubín Ramírez

Instituto Tecnológico de Tepic, México
nrubin@ittepic.edu.mx

Recepción: 5/noviembre/2019

Aceptación: 17/diciembre/2019

Resumen

El presente artículo se deriva de la implementación de un algoritmo de Redes Neuronales, como un acercamiento de la industria 4.0, que busca obtener la predicción de un rango de valores en el cual los parámetros permitan proporcionar un ahorro de scrap de aluminio en las inyectoras de fundición. Con la implementación del algoritmo, al realizar la predicción de los parámetros óptimos, se logró disminuir en un 25% en promedio los defectos de poros, mallenado, falta material y fisuras, así como el reproceso de las piezas de aluminio. El software instalado permite, además, contar con un histórico en tiempo real de cada turno de los operadores de las máquinas inyectoras. Después de haber realizado pruebas en las máquinas en donde se instaló el sistema, se pudo apreciar la gran utilidad que representó el uso de inteligencia artificial para predecir y hacer eficientes los procesos de las inyectoras de aluminio, al haber reducido la cantidad de scrap y defectos en las piezas de aluminio.

Palabras Clave: Fundición, Industria 4.0, red neuronal artificial, scrap, predicción.

Abstract

This article is derived from the implementation of an algorithm of Neural Networks, as an approach of industry 4.0, which seeks to obtain the prediction of a range of values in which the parameters allow to provide savings of aluminum scrap in the injectors of foundry. With the implementation of the algorithm, when predicting the optimal parameters, it was possible to reduce pore defects, filling, material and cracks by 25% on average, as well as the reprocessing of aluminum parts. The installed software also allows you to have a real-time history of each shift of the operators of the injection machines. After having carried out tests on the machines where the system was installed, it was possible to appreciate the great usefulness of the use of artificial intelligence to predict and make efficient the processes of aluminum injectors, having reduced the amount of scrap and defects in the aluminum parts.

Keywords: Foundry, Industry 4.0, artificial neural network, scrap, prediction.

1. Introducción

El tema de Inteligencia Artificial (IA) en la industria 4.0 está experimentando una rápida evolución, sobre todo en las empresas manufactureras donde se busca automatizar y controlar los sistemas de instrumentación, PLC's (*Programmable Logic Controllers*), CNC's (*Computer Numerical Control*), robots industriales, etc., dichos instrumentos están integrados dentro del sistema de información de la industria manufacturera, generando grandes cantidades de datos que deben ser manipulados para ser convertidos en información significativa para automatizar los procesos de control.

El software sigue siendo una herramienta elemental en las empresas manufactureras. Freeman y Skapura [1993], describen la importancia que tiene el uso del software en las nuevas metodologías informáticas necesarias para los desarrollos que se requieren, todo esto también relacionado con los nuevos requerimientos y sistemas abiertos de estandarización. Kagermann, Lukas y Wahlster [2011], mencionan que la IA, tiene un papel muy importante en todo lo que representa la industria 4.0 debido a las técnicas que provee, como son:

sistemas expertos, aprendizaje automático, robótica, sistemas inteligentes y lógica borrosa, entre otras.

El principal reto a atender en la empresa en donde se hizo este trabajo de investigación, fue la reducción de scrap de aluminio en las inyectoras de fundición, reducir los defectos de poros, mallenado, falta material y fisuras, así como el reproceso de las piezas de aluminio. Para resolver esta problemática se implementó un algoritmo para la clasificación de datos relacionados con los valores de parámetros de diversos moldes. Con las bases de datos que se generan, posteriormente se analizan y se estiman los rangos en los que los parámetros relacionados a cada molde son los adecuados para eliminar los defectos anteriormente mencionados en el proceso de fundición.

El objetivo de este trabajo es implementar un algoritmo de predicción que permita, mediante el uso de Redes Neuronales, la disminución de defectos residuales y obtener de esta manera, una mejoría en la calidad de producción de piezas de aluminio.

Concepto de Industria 4.0

El término industria 4.0, fue por primera vez acuñado en Alemania en el año 2011, donde a nivel industrial se hace referencia a la transformación digital con el cambio del internet tradicional a internet de las cosas (IOT), el cual es el principal componente de la industria 4.0. El IOT ha buscado puntos de oportunidad para ser utilizado a nivel industrial, con la finalidad de tener conectividad en tiempo real, por la convergencia que existe con las TICs, y con el uso de la robótica, también considerada como parte medular de algunas industrias que son utilizadas para la automatización de sus procesos. En este trabajo se hará evidente el uso del IOT, la conectividad en tiempo real, así como el uso de las TICs, todo ello dentro de los elementos que conforman la industria 4.0.

Redes Neuronales Artificiales

Uno de los elementos a utilizar en este trabajo para el procesamiento de datos es el uso de las Redes Neuronales. Para Freman y Skapura [1993], una red

neuronal es un sistema de procesadores paralelos conectados entre sí en forma de grafo dirigido. Su esquema de cada elemento de procesamiento denominado 'neuronas' de la red, se presenta como un nodo. Dichas conexiones establecen una estructura jerárquica tratando de emular la fisiología del cerebro humano y buscar nuevos modelos de procesamiento para solucionar problemas concretos del mundo real. Por la versatilidad de las RNA y debido a su comportamiento para aprender, reconocer y aplicar relaciones entre objetos y tramas de objetos propios de la vida cotidiana, son utilizadas como herramienta que puede llegar a resolver problemas difíciles, tanto de clasificación como de reconocimiento. Dicha posibilidad de resolución de problemas difíciles, se debe a los principios de las redes neuronales, que cuentan con las siguientes características que cita Hilera y Martínez [1995].

Características de las Redes Neuronales

Una característica importante de las RNA es el aprendizaje adaptativo, debido al comportamiento de entrenamiento con una serie de ejemplos ilustrativos, de esta forma no es necesario elaborar un modelo a priori, ni el establecimiento de funciones probabilísticas. Una red neuronal es adaptativa, si puede modificarse en repetidas ocasiones con el fin de obtener nuevas condiciones de trabajo.

Mientras que el aprendizaje es un proceso donde se modifica información interna de la red neuronal artificial, la autoorganización en una red consiste en la modificación de la red completa con el fin de llevar a cabo un objetivo específico. Autoorganización significa generalización, de esta forma una red puede responder a datos o situaciones que no ha experimentado antes, pero que puede inferir sobre la base de su entrenamiento. Esta característica es muy útil sobre todo cuando la información de entrada es poco clara o se encuentra incompleta.

Una característica importante de las redes neuronales es que son tolerantes a los fallos, la razón es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento. La mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos almacenan cada pieza de información en un espacio único, direccional y

localizado, en cambio, las redes neuronales almacenan información no localizada, de ahí que la mayoría de las interconexiones entre los nodos de la red tendrán sus valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada.

Una de las mayores prioridades, casi en la totalidad de las áreas de aplicación, es la necesidad de realizar procesos con datos de forma muy rápida. Las redes neuronales se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las redes puedan operar en un entorno de tiempo real, la necesidad de cambio en los pesos de las conexiones o entrenamiento es mínimo. Para profundizar un poco en el tema de las RNA, se puede mencionar que el término neurona no es de extrañar que se piense que uno se esté refiriendo al cerebro humano, y esto es debido a que las RNA están inspiradas biológicamente. Se dice que el hombre posee cerca de diez millones de neuronas masivamente interconectadas y que una neurona es una célula que está especializada para propagar una señal electroquímica.

La estructura ramificada de las neuronas en cuanto a las entradas (dendritas en el cerebro humano) y la estructura ramificada de salida (axones en el cerebro humano), se conectan entre sí, es decir, los axones de una célula se concretan con las dendritas de otra por vía de la sinapsis, de ahí que el principio de las RNA sea imitar el funcionamiento cerebral.

Estructura de las RNA

Las RNA están formadas por una gran cantidad de neuronas que suelen denominarse nodos o unidades procesadoras. Un nodo o neurona cuenta con una cantidad variable de entradas (X_1, X_2, \dots, X_n) y a su vez dispone de una sola salida, la cual transmite la información al exterior o hacia otras neuronas. Cada salida o señal de salida tiene asociada un peso que se calculará en función de las entradas, por lo que cada una de ellas es afectada por un determinado peso definido por w_i , con $i=1,2,\dots,n, n+1$, que modifica las entradas antes de que sean sumadas y suministradas al elemento de umbral. En este sentido, los pesos son similares a las sinapsis en el sistema neuronal humano. La función que transforma la salida

correspondiente a la suma en la salida final, se denomina *función de activación* y toma el valor de 1 si su argumento es verdadero y -1 si es falso, como se puede ver en la figura 1 [Pajares y Santos, 2006].

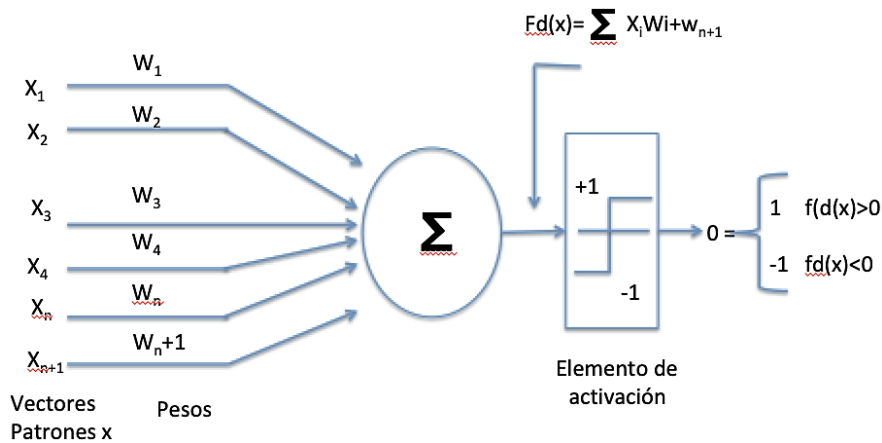


Figura 1 Modelo de perceptrón.

El proceso de aprendizaje consiste en hallar los pesos que codifican los conocimientos. Una regla de aprendizaje hace variar el valor de los pesos de una red hasta que éstos adoptan un valor constante, cuando esto ocurre se dice que la red ya "ha aprendido".

Cuando se conectan varias neuronas, se consigue una red que puede presentar diversas variaciones en las tipologías, las cuales son clasificadas según los criterios que se exponen a continuación:

- Número de niveles o capas.
- Número de neuronas por nivel.
- Formas de conexión.

De acuerdo al problema que se vaya a solucionar, el diseño de una u otra tipología variará. Si tomamos como ejemplo la elaboración de un programa de filtro digital en una computadora, se deberá emplear un algoritmo que permita que todas las capas estén uniformemente interconectadas, es decir, que todos los nodos de una capa estén conectados con los nodos de otra capa. Para hacer referencia a este tipo de red, la figura 2 muestra la arquitectura de una RNA clásica con variables

de entrada, dos capas intermedias y una capa de salida, todas ellas conectadas entre sí.

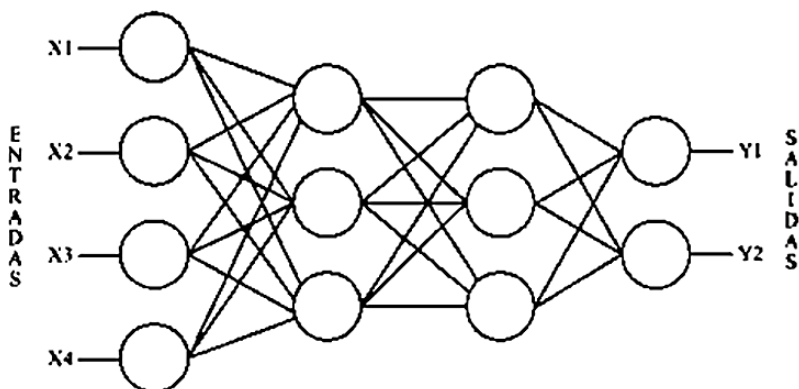


Figura 2 Esquema de una RNA de dos capas de neuronas intermedias.

Existe una gran variedad de modelos de RNA que dependen del objetivo para los cuales fueron diseñados y del problema práctico a solucionar. En la actualidad, el éxito de las RNA depende muchas veces del usuario, para lograr alcanzar aplicaciones que sean de gran utilidad y porque no, hasta valor comercial. Sin embargo, hay algunos modelos más populares que otros, uno de estos modelos es referenciado por el autor Pao YH [1995], quien menciona que el modelo Adeline/Madaline fue diseñado por Bernard Widrow [1956], y es utilizado en el diseño y realización de filtros, así como en la eliminación de ruido en señales portadoras de información, como es el caso de los módems. El modelo backpropagation ha sido utilizado en proyectos del tipo neural fonética, mientras que el modelo de los mapas auto-organizados (Self-Organizing Map, SOM), mejor conocido como modelo de Kohonen y el perceptrón simple, ambos se utilizan con buen porcentaje de éxito en la clasificación de clases.

2. Metodología

La metodología que se empleó en este proyecto para el análisis de datos se puede describir en general de la siguiente manera:

- Instrumentar dispositivos de medición en las máquinas de fundición para la obtención de datos.

- Generar algoritmos para la obtención de datos. Se establecieron los tiempos de corte adecuados para la obtención de datos en función de los turnos de trabajo y del molde empleado.
- Se desarrollaron los algoritmos para el análisis de datos apoyados con el uso del software Matlab, para la predicción de los parámetros de medición adecuados que permitan un proceso de fundición de piezas de aluminio con calidad y menor cantidad de scrap.
- Se diseñaron tablas de salida para la correcta entrega de reporte de datos.

En la figura 3 se muestra un diagrama de la metodología expuesta:

- **Fase 1.** La obtención de los datos se relacionó con la información que las inyectoras arrojaron, estos datos se obtuvieron en tiempo real y se realizó una normalización para analizarlos, y todo dependió del tipo de PLCs, ya que cada una de las máquinas arrojan diferentes rangos de parámetros.

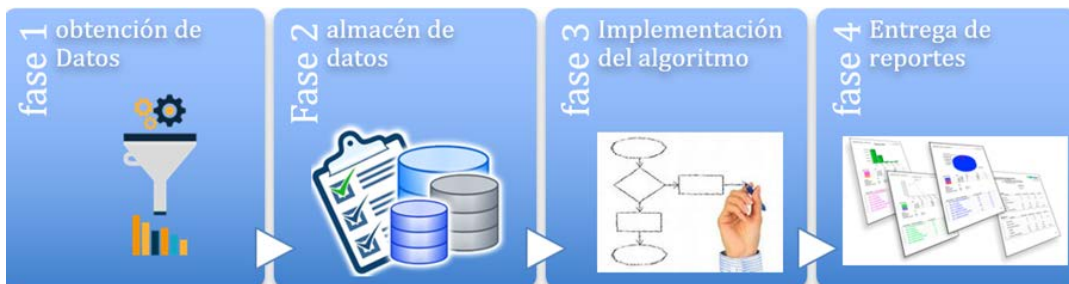


Figura 3 Fases para el desarrollo de la implementación del algoritmo de RNA.

Para comprender un poco más el proceso de adquisición de datos, se describe a grandes rasgos la visión de este procedimiento:

La planta se encuentra constituida por varias células que constan de una serie de maquinarias y elementos como: tina de enfriamiento, contenedor de scrap, rampas, horno de alimentación, robots y las inyectoras, las cuales cuentan con diversas tecnologías en diferentes versiones, por lo que se dividieron en cuatro familias como se muestra en la tabla 1.

Las familias 1, 2 y 3 fueron aquellas en las que a pesar de que la obtención de datos se realizó en diversas formas, fue posible. Mientras que en la

familia 4 están aquellas inyectoras en las cuales fue imposible obtener los datos debido a tecnología obsoleta con la que cuentan.

Tabla 1 Familias de Inyectoras (Las inyectoras 11 y 15 son las inyectoras piloto).

Familia 1	Familia 2	Familia 3	Familia 4
Iny. 15	Iny. 06	Iny. 11	Iny. 16
Iny. 14	Iny. 05	Iny. 12	Iny. 13
Iny. 01	Iny. 04	Iny. 19	Iny 18.
Iny. 10	Iny. 07	Iny. 20	
Iny, 17	Iny. 08	Iny. 21	
	Iny. 09	Iny. 22	
		Iny. 23	
		Iny. 24	
		Iny. 25	

Para obtener los datos en tiempo real en las inyectoras de las familias 1 y 3 se agregó a cada una de las inyectoras a una sola red (sin salida a Internet) asignándoles IPs estáticas, de acuerdo con las políticas de la empresa. Cualquier sistema conectado a una red distinta no puede realizar una conexión directa con otra red (aunque ambas pertenezcan a la empresa). Para la familia 1 fue necesario asignar una excepción de entrada para el puerto 3306 de MySQL para poder realizar la conexión a las inyectoras sin necesidad de desactivar el firewall. Cada familia tiene diversas maneras de adquirir los datos, los cuales se muestran en los puntos siguientes:

- ✓ **Familia 1:** está constituida por cinco inyectoras Pretransa con Windows XP en donde se tiene el manejador de base de datos MySQL en versiones distintas, en esta familia los registros de los parámetros son almacenados en la base de datos la cual para todas las inyectoras dentro de la familia 1 se llaman de la misma manera “pci” y las tablas comparten de igual manera el mismo nombre. Para la obtención de los datos de la Familia 1 se realizó una consulta mediante un “Query” o consulta a la base de datos, para lograrlo se necesitó crear diversos orígenes de datos para diferenciar cada una de las inyectoras estableciendo la IPs estáticas, como se observa en la figura 4.

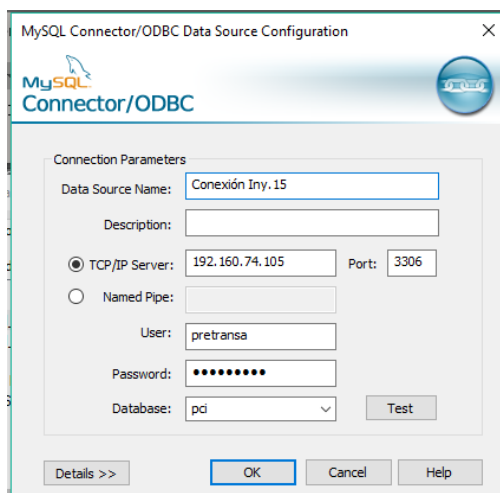


Figura 4 Creación de un origen de datos.

Cuando se procedió con lo anterior, se realizó el código necesario para la obtención de los datos, esta adquisición se realizó en un intervalo de 10 minutos ya que una inyectora consigue realizar una pieza en un rango de entre 40 y 60 segundos aproximadamente.

- ✓ **Familia 2:** en esta familia se tienen seis inyectoras las cuales de igual manera que la familia 1 tienen Windows XP pero una gran desventaja que se presentó fue que no contaban con función de red en la estructura interna de las inyectoras por lo que existió la necesidad de adquirir la información desde una base de datos remota, además de que los datos que genera cada una de las inyecciones se ingresa a un archivo de texto, que, al momento de ser extraído de la inyectora se cifra y actualmente no se cuenta con la información necesaria para aplicar un algoritmo de descifrado, y la recolección de los datos de esta familia sería sumamente complicada ya que se deberían adquirir de manera física como por ejemplo: bajar la información a una memoria tipo USB, corriendo el peligro de en algún momento infectar al sistema de la inyectora con alguna clase de virus. Por lo que se obtendrá la información de la base de datos remota que no es del todo verídica ya que los operadores son los encargados de capturar esa información en el sistema que se encuentra en línea, pero al

contar con tecnología antigua en estas inyectoras no se cuenta con los controladores (drivers) necesarios para establecer este tipo de función por lo que se eligió esta opción como la más adecuada para adquirir los datos necesarios para aplicar la predicción en estas seis inyectoras.

- ✓ **Familia 3:** está constituida por nueve inyectoras con el sistema operativo Windows XP. En esta familia se obtuvo la información por medio de un programa que se desarrolló para la lectura de un archivo tipo XML y la obtención de la información de los parámetros a evaluar. Esta adquisición se realiza cada cierto periodo de tiempo ya que el archivo en el que se registra la información de las inyecciones se sobrescribe. La justificación para la sobreescritura del archivo se debe a la poca capacidad de almacenamiento que se tiene en el sistema de las inyectoras de esta familia.

Otra parte importante en la adquisición de datos, es que a la par de la información de rangos de parámetros, también se obtuvo la cantidad de defectos que se producían después de que se ha hecho la inyección de piezas y el maquinado. Estas variables registradas por los “monitoristas”, se obtienen cada hora (cada hora se toman 10 piezas y se evalúan. Con el resultado se puede obtener la cantidad de cada defecto), las variables son:

- ✓ Cantidad de piezas con falta de material.
- ✓ Cantidad de piezas con “mallenado”.
- ✓ Cantidad de piezas con poros.
- ✓ Cantidad de piezas con fisuras.
- ✓ Cantidad de piezas con otro tipo de defecto.

Además, se obtienen variables registradas por el personal técnico, las cuales se adquieren cada fin de turno y estas son las siguientes:

- ✓ Temperatura del aluminio.
- ✓ Valor de los caudalímetros.
- ✓ Temperatura del horno.
- ✓ Tiempo de ciclo.

- **Fase 2.** Una vez obtenida la información anterior, se almacenó en una base de datos remota, la cual se manipula para su normalización. Esto permite evitar tener errores y redundancia de la información almacenada. Posteriormente se introduce la información en los algoritmos de predicción que se emplearán para el sistema experto.

El almacenamiento de datos se realiza cuando se lleva a cabo la adquisición de los datos, por lo que puede dividirse en dos formas:

- ✓ De manera automática, mediante un *query* de *insert* se ingresa un registro a la base de datos cada cierto tiempo. Por hora se obtienen alrededor de 85 datos, es decir, cada 42 o 45 segundos se inyecta una nueva pieza. Esto procede para todas las familias a pesar de las diversas tecnologías empleadas.
- ✓ Por inserción en la base de datos que contendrá el histórico (figura 5).

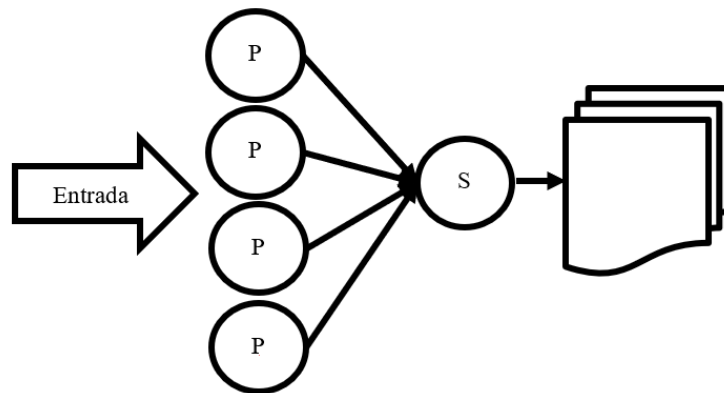


Figura 5 Modelo propuesto para la estructura del algoritmo RNA.

- **Fase 3.** Implementación del algoritmo. El algoritmo está sustentado en técnicas de inteligencia artificial como lo es el Aprendizaje Automático, el cual está basado en redes neuronales (mostradas en la figura 2), lo cual permite realizar una optimización de los datos y permite obtener una buena toma de decisiones en cuanto a los procesos de la empresa.
- **Fase 4.** En esta fase se realizan los reportes correspondientes, los cuales pueden ser visualizados con hojas de cálculo y gráficas. Además, se pueden

visualizar en pantallas táctiles y los operadores puedan manipular y visualizar los datos en línea.

3. Resultados

En este apartado, se mostrarán los resultados derivados de la captación de los datos que se obtuvieron por parte de las máquinas del área de fundición. Es importante mencionar que los datos solo aquellos que están relacionados con una máquina, y se fueron recolectando cada 10 minutos, por hora y por jornada, la cual consta de una jornada normal de 8 horas. La tabla 2 presenta una muestra de las variables del proceso para alimentar el sistema experto, así como algunos datos de una de las corridas que se hicieron en la inyectora.

Tabla 2 Parámetros obtenidos de la máquina inyectora T.

IdCurva	V1	P1	H2	V2	H3	T2	P3	T3	HC	FECHA	HORA	DIF
146511	0,16	118	278	3,9	400	38	249	0	48	08/11/2017	16:24:03	00:00:42
146510	0,17	117	279	3,92	402	38	247	0	47	08/11/2017	16:23:20	00:00:43
146509	0,16	116	277	3,9	401	40	249	0	47	08/11/2017	16:22:36	00:00:44
146508	0,19	117	284	3,89	399	34	246	0	45	08/11/2017	16:21:52	00:00:42
146507	0,17	117	278	3,91	398	37	245	0	46	08/11/2017	16:21:09	00:00:44
146506	0,2	118	285	3,93	402	34	243	0	47	08/11/2017	16:20:25	00:00:44
146505	0,17	117	279	3,9	400	38	245	0	43	08/11/2017	16:19:41	00:00:42
146504	0,19	117	284	3,92	404	35	244	0	44	08/11/2017	16:18:58	00:00:44
146503	0,16	117	277	3,91	405	40	246	0	44	08/11/2017	16:18:14	00:00:44
146502	0,17	117	280	3,92	403	38	241	0	47	08/11/2017	16:17:30	00:00:42
146501	0,16	118	277	3,87	400	40	244	0	46	08/11/2017	16:16:47	00:00:43
146500	0,18	117	281	3,82	404	38	246	0	46	08/11/2017	16:16:03	00:00:44
146499	0,16	118	278	3,72	399	39	250	0	45	08/11/2017	16:15:19	00:00:42
146498	0,18	117	280	3,9	401	36	244	0	45	08/11/2017	16:14:36	00:00:44
146497	0,16	118	278	3,9	402	39	249	0	46	08/11/2017	16:13:52	00:00:43

Una vez recolectados los resultados, se envían a una interfaz a nivel usuario, la cual muestra la manera de informar la predicción que arroja la red neuronal de tipo backpropagation. Esta información se generó previamente en el análisis de datos para que se visualizaran los parámetros adecuados para la toma de decisiones, como se muestra en la tabla 3.

Tabla 3 Valores de parámetros iniciales para un molde.

	V1	P1	H2	V2	H3	T2	P3	HC
Máximo	0.4500	150	330	4.5000	450	500	330	50
Nominal	0.2500	82.5000	300	4	400	250	300	30
Mínimo	0.0500	15	270	3.5000	350	0	270	10

Se puede observar que el usuario podrá ver los datos por cada uno de los parámetros, visualizando los mínimos, máximos y valores nominales de cada uno de ellos. esto permitió que se generara una gráfica en la que se muestra cuando se tienen que tomar decisiones para hacer un cambio de molde. Estas alertas están determinadas como rojo para valores críticos, lo cual indica que se encuentra fuera del rango del límite del valor de los parámetros. El color amarillo alerta que se encuentra dentro del rango de valores para los parámetros, pero no está en el rango central más óptimo y el color verde avisa que los valores se encuentran dentro del rango permitido y representan los valores más óptimos.

En la figura 6 se muestra la gráfica del comportamiento de los datos para las corridas anteriormente mencionadas, y se puede observar cuál de ellos está cayendo en cada una de las alertas. Mientras el valor de los parámetros se encuentre dentro del área verde o amarilla, se considerará que el rango aún es aceptable y el sistema experto deberá modificar el valor del rango reduciéndolo.

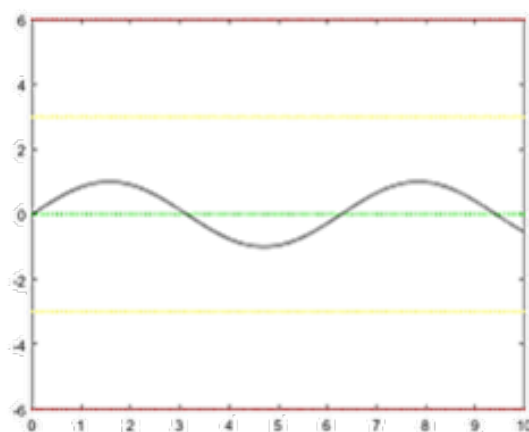


Figura 6 Gráfica de alertas visuales del estado de los parámetros.

En cuanto a la red neuronal, inicialmente se utilizó perceptrón simple pero debido a que la red no aprendía de manera adecuada, se cambió al uso del perceptrón multicapa como modelo de neurona o función de transferencia “purelin”, aunque también se pueden usar las funciones “logsig” y “tansig”. Estas funciones son las más utilizadas en redes multicapa. En la la figura 7 se puede observar como la función “purelin” es una función de transferencia lineal.

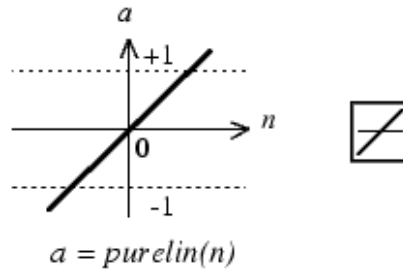


Figura 7 Gráfica de la función de transferencia "Purelin".

Cuando se entrena una red neuronal multicapa, la información se divide en tres subconjuntos, el primero es el subconjunto de entrenamiento el cual se utiliza para calcular el gradiente y actualizar los pesos de la red y las vías. El siguiente es un conjunto de validación, el error de validación se monitorea durante el entrenamiento y por último el conjunto de prueba, el cual no es usado durante el entrenamiento, pero es útil al comparar diversos modelos.

Como algoritmo de entrenamiento se utilizó el de "Levenberg-Marquardt" que no es más que dar solución mediante mínimos cuadrados ponderados, siendo éste el algoritmo generalmente más rápido. Durante el entrenamiento, el proceso puede ser observado en la ventana de entrenamiento como se ve en la figura 8.

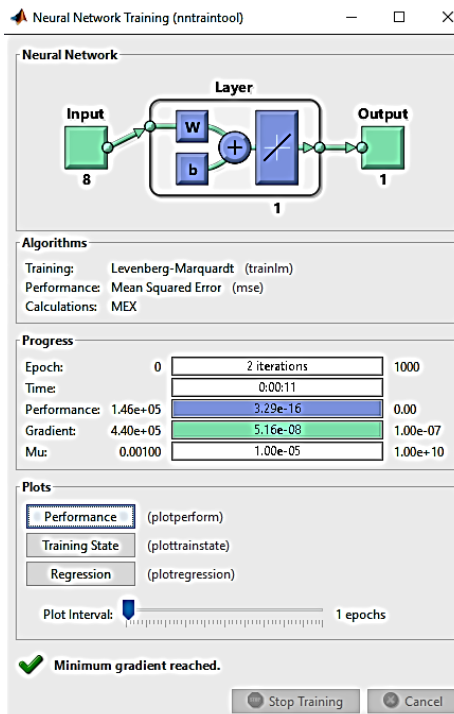


Figura 8 Ventana de entrenamiento de la red neuronal.

En la figura 9 se puede observar que la predicción se realizó en dos épocas, que es el número de iteraciones en la red neuronal para llegar al resultado, además, muestra un tiempo máximo de entrenamiento de 11 segundos. En la ventana de entrenamiento, se pueden advertir tres botones, uno de ellos permite observar el desempeño o resultado de la predicción de la red (botón “Performance”).

El estado entrenamiento (botón “Training state”) muestra el progreso de otras variables de entrenamiento, como la magnitud del gradiente, el número de checks de validación, entre otros, como se observa en la figura 10.

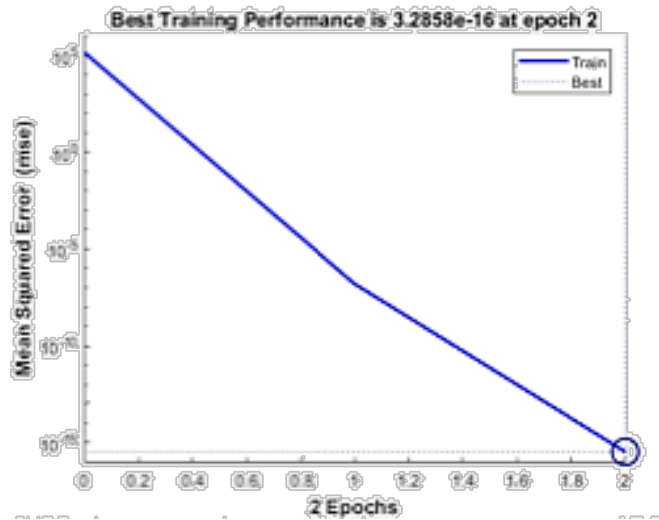


Figura 9 Gráfica del desempeño de la red

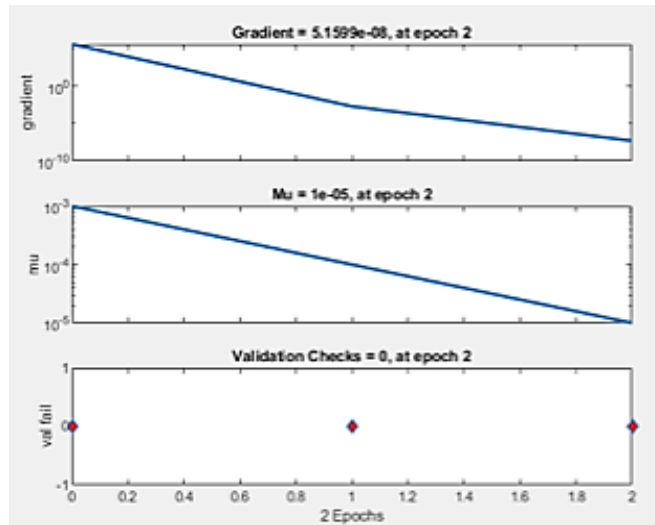


Figura 10 Gráfica del estado de entrenamiento.

Por último, se encuentra el botón de “Regression” el cual se visualiza una gráfica que muestra una regresión entre las salidas de la red y los objetivos de la red, que solo se tomaron algunos de forma demostrativa, se presentan en la figura 11.

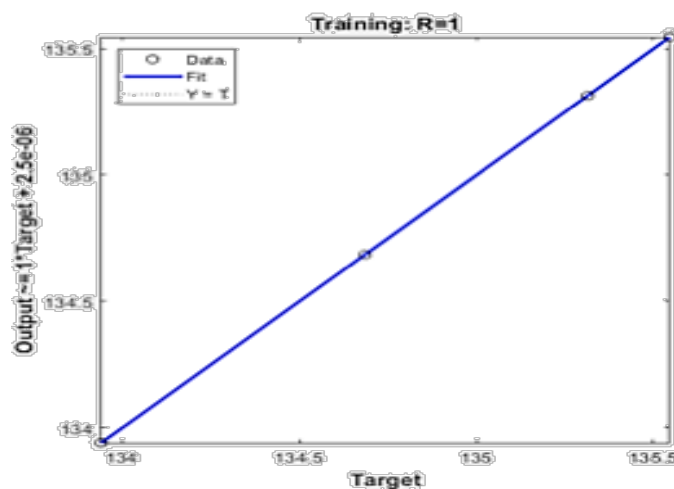


Figura 11 Gráfica de Regresión en el resultado de la predicción.

4. Conclusiones

Dentro de la fase de adquisición de datos se encontraron algunos conflictos con las conexiones que se deberían establecer a las inyectoras debido a las restricciones que se tenían en cuanto a conexiones remotas, por lo que se requirió el apoyo del personal de la empresa (como supervisores, técnicos, monitoristas, personal de mantenimiento, entre otros) para hacer posible el logro de estas adquisiciones de datos. Sin embargo, se considera que es posible tomar como oportunidad de mejora esta parte para facilitar el trabajo de los técnicos, ya que aún es necesario que recaben información de manera manual.

En el desarrollo de la implementación del algoritmo, se requirió mantener un contacto estrecho con el experto en el área, para tener el conocimiento apropiado de lo que se debía implementar, ya que de manera constante existieron cambios. Sin embargo, a pesar de que existieron algunos contratiempos en el desarrollo del proyecto, se logró recolectar la información para normalizar los datos y alimentar a la base a un servidor remoto que permitió analizar la información para obtener la predicción de los parámetros óptimos que permitieron eliminar 25% del scrap.

La reducción de fallas se logró a través de la implementación del algoritmo de la red neuronal, el cual permitió la predicción de los parámetros reales de las máquinas, en comparación con los datos de diseño de la configuración original referidos al desgaste de los moldes y fallas de las piezas de fundición de aluminio.

5. Bibliografía y Referencias

- [1] Freeman, J. A., Skapura, D. M. (1993). *Redes Neuronales. Algoritmos, técnicas de propagación*. México: Addison-Wesley.
- [2] Fuente, E. (2015). *Industria 4.0*. *Dyna Ingenieria e Industria* 90(3): 16–17.
- [3] Hilera J., Martínez V. (1995). *Redes neuronales: fundamentos, modelos y aplicaciones*. Madrid, RA-MA,
- [4] Kagermann, H., Lukas, W., Wahlster, W. (2011). *Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution*. En *Ingenieur.de*: <https://www.ingenieur.de/technik/fachbereiche/produktion/industrie-40-mit-internet-dinge-weg-4-industriellen-revolution/>.
- [5] Kohonen, T. (2017). *Self-Organizing-Maps*. Springer, Berlin, Second edition., 362 p.
- [6] Pajares, M., Santos, M. (2006). *Inteligencia artificial e Ingeniería del Madrid*, RA-MA.
- [7] Pao Y.H., (1995). *Neural Net Computing for pattern Recogniton*. In Chen, L.F. Pau and P.S.P. Wang (Eds) *Handbook of Pattern Recogniton and Computer Vision*, 125-162 p.
- [8] Zegarra, C. y Pérez, M. (2018). *Industria 4.0: oportunidades y retos en México*. En revista *Forbes*: <https://www.forbes.com.mx/industria-4-0-oportunidades-y-retos-en-mexico/>.