ISSN 1870-4069

Sistema de detección de intrusos en el CAN bus de automóviles utilizando una red neuronal multicapa

Juan Carlos Venegas-Segura[1], Ponciano Jorge Escamilla-Ambrosio[1], Alberto Jorge Rosales-Silva[2], Floriberto Ortiz-Rodríguez[2], Narciso Muñoz-Aguirre[3]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, México

[2] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, México

[3] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Azcapotzalco, México

Karlvsk16@gmail.com,{pescamillaa,arosaless, flortiz, nmunoz}@ipn.mx

Resumen. La constante evolución tecnológica en los vehículos ha traído consigo un sin número de ventajas permitiéndole brindar mejores servicios. No obstante, se ha demostrado en múltiples investigaciones que los vehículos son susceptibles a sufrir ataques de ciberseguridad. En consecuencia, en este trabajo se propone un sistema de detección de intrusos en el CAN bus de un automóvil utilizando una red neuronal multicapa. Se realizó un proceso de experimentación para encontrar la estructura de la red neuronal que realice de manera satisfactoria la tarea en cuestión. Esto incluyó experimentar con el número de capas ocultas y el número de neuronas en dichas capas. La red neuronal resultante se encarga de detectar tráfico malicioso en el CAN bus, permitiendo identificar datos libres de ataque y datos con algún tipo de ataque. Los ataques que se consideran en este trabajo corresponden a los ataques tipo denegación de servicio, suplantación de identidad e inyección de datos aleatorios. Se demuestra de manera experimental que, una vez encontrada la estructura adecuada y entrenada la red neuronal propuesta, ésta es capaz de identificar los ataques antes mencionados.

Palabras clave: Ciberseguridad, sistema de detección de intrusos (IDS), red neuronal multicapa, CAN bus, electronic control unit (ECU).

Intrusion Detection System in the CAN bus of Automobiles Using a Multilayer Neural Network

Abstract. The constant technological evolution in vehicles has brought with it several advantages that allow it to provide better services. However, it has been shown in multiple investigations that these vehicles are susceptible to

cybersecurity attacks. Consequently, this work proposes an intrusion detection system in the CAN bus of automobiles using a multilayer neural network that is responsible of protecting this type of networks. An experimental process was carried out in order to find the structure of the neural network to obtain satisfactory results for the task at hand. This included experimentation with the number of hidden layers and the number of neurons on such layers. The neural network found detects malicious traffic on the CAN bus, allowing to discriminate free-attack data and data with some type of attack. The attacks considered in this work correspond to denial of service, impersonation and fuzzy. It is shown that, once the neural network is trained, it can identify the attacks previously mentioned.

Keywords: Cybersecurity, intrusion detection system (IDS), multilayer neural network, controller area network (CAN), electronic control unit (ECU).

1. Introducción

El uso de tecnología en los vehículos ha mejorado muchos procesos en los cuales los dispositivos mecánicos utilizados tenían cierta desventaja debido al tiempo de respuesta que necesitaban o incluso al desgaste que sufrían, lo que en ocasiones provocaba algunas descomposturas.

Con la intervención de la electrónica y la introducción de computadoras en los vehículos, que en este contexto se conocen como ECU (Electronic Control Unit, en inglés), se logró un avance tecnológico en la industria automotriz. La aparición del CAN bus (del inglés, Controller Area Network) permitió reducir la cantidad de cableado que se requería para mantener conectado cada uno de los dispositivos dentro de un vehículo.

Hoy en día, el CAN bus se ha convertido en el principal protocolo de comunicación en cualquier vehículo. Sin embargo, múltiples investigaciones como las realizadas por los hackers Charli Miller y Chris Valasek [1], demuestran que tienen una gran cantidad de vulnerabilidades, permitiéndole a un atacante ocasionar algún mal funcionamiento o incluso tomar el control de algún sistema del vehículo.

En este trabajo se presenta una investigación relacionada a los problemas de ciberseguridad que se tienen en el CAN bus de los automóviles actuales. Se presenta una solución que utiliza una red neuronal multicapa para la detección de tres tipos de ataque, además del estado sin ataque. Los ataques que se consideran en este trabajo corresponden a los ataques tipo denegación de servicio (DoS, Denial of Service, en inglés), suplantación de identidad (impersonation, en inglés) e inyección de datos aleatorios (fuzzy attack). El presente trabajo se encuentra estructurado de la siguiente manera.

En la Sección 2 se presenta un resumen de los trabajos relacionados. En la Sección 3 y Sección 4 se presentan algunas consideraciones técnicas que fueron tomadas en el desarrollo de este trabajo.

Finalmente, en la Sección 5 se presenta el proceso que se siguió para encontrar una estructura adecuada de una red neuronal que permita dar solución al problema planteado, donde se determinó el número de capas ocultas, así como el número de neuronas que tendrá cada una de ellas.

2. Trabajos relacionados

Hoy en día, existen muchos trabajos de investigación que proponen un sistema de seguridad para la protección de la red de un vehículo. Muchas de estas investigaciones se enfocan en el intercambio de información a través del CAN bus, en especial la frecuencia con la que cada computadora en un vehículo (ECU) envía mensajes a través de éste. Se ha demostrado que cada ECU envía mensajes de forma periódica y, por lo tanto, esta característica se verá alterada durante un ataque.

Centrándose en trabajos que proponen un sistema de detección de intrusos (IDS, Intrusion Detection System, en inglés) usando herramientas del área de aprendizaje de maquina (Machine Learning, en inglés) como son las redes neuronales, se encuentra el trabajo realizado por Hyung Ming Song et al. [2] quienes proponen un IDS basado en una red convolucional profunda (Deep Convolutional Neural Network, en inglés) reportando una exactitud del 99%. Min-Joo Kang et al. [3] proponen un IDS usando una red neuronal profunda (Deep Neural Network, en inglés) reportando una exactitud del 98 %. Por su parte Hyung Ming Song et al. [4] proponen un IDS basado en una Red GAN (Generative Adversarial Networks, en inglés) y reportan una exactitud del 98 %.

A diferencia de los trabajos anteriores, en la presente investigación se realiza un proceso de experimentación en el cual se determina el número de capas ocultas y el número de neuronas que tendrá cada una de ellas. Esto se realiza con el objetivo de encontrar la estructura de una red neuronal lo más simple posible con el fin de reducir la complejidad de la RN, pero que resuelva el problema en cuestión sin la necesidad de emplear una red neuronal compleja como las reportadas en la literatura. Asimismo, se utilizan los datos de entrada en un formato de ventana deslizante lo que permitirá llevar a cabo un proceso de clasificación más rápido y en tiempo real.

3. CAN bus

El CAN bus [16] es una representación de un bus serial diseñado en 1986 por la compañía alemana Robert Bosh, para su uso en el desarrollo industrial y su aplicación en vehículos. El protocolo CAN está basado en una topología bus para la transmisión de mensajes en tiempo real y entornos distribuidos y que ofrece una gestión a la comunicación entre múltiples unidades de control.

El protocolo CAN soporta frecuencias de datos de hasta de 125 kbps, con una distancia de hasta 40m, el cual provee una mayor capacidad de transferencia de datos y rango extendido de transmisión, Además, el CAN puede soportar tasas de transferencia de hasta 1 Mbps.

El protocolo CAN contempla cuatro tipos de mensajes: mensaje con datos (data frame), mensaje de solicitud remota (remote frame), mensaje de error (error frame) y mensaje de sobrecarga (overload frame).

La arquitectura del CAN bus permite llevar a cabo el proceso de comunicación entre las diferentes unidades de control, permitiéndoles identificar a cada nodo (a cada ECU) solo aquellos mensajes que le pudieran servir.

Desafortunadamente, el protocolo CAN fue diseñado sin contar con algún sistema de protección, por lo que cada mensaje que circula por el mismo es tomado como verídico. En consecuencia, si algún mensaje falso es procesado por alguna ECU, se

llevará a cabo alguna acción que pudiera provocar alguna avería, un mal funcionamiento o incluso alguna situación catastrófica**.**

4. Redes neuronales

En general no existe una sola definición de red neuronal, y ésta puede variar de acuerdo con el texto o al artículo consultado. Sin embargo, a grandes rasgos se puede decir que una red neuronal es un modelo computacional compuesto de unidades procesadoras que se encuentran interconectadas entre sí. Estos sistemas buscan emular el comportamiento del cerebro humano con el fin de solucionar múltiples problemas, uno de ellos es el de clasificación.

En el presente trabajo se plantea dar solución a los problemas de ciberseguridad en el CAN bus que tienen los vehículos hoy en día. Como se ha mencionado antes, es posible que algún agente atacante pueda realizar diversos ataques a una red vehicular permitiéndoles llevar a cabo diferentes tareas.

Se plantea utilizar una red neuronal multicapa que permita dar solución al problema antes planteado. El proceso para desarrollar la red neuronal será a través de experimentos que permitan encontrar las principales características que conforman a la red neuronal: optimizadores, datos de entrada, número de capas ocultas, número de neuronas en cada capa intermedia, entre otros**.**

4.1. Arquitectura de una red neuronal

Se denomina arquitectura a la topología o estructura de una red neuronal. En una red neuronal artificial los nodos se conectan por medio de sinapsis, estando el comportamiento de la red determinado por la estructura de conexiones sinápticas. En general los nodos (neuronas) se suelen agrupar en unidades estructurales que se denominan capas. El conjunto de una o más capas constituye la red neuronal.

Se distinguen tres tipos de capas: de entrada, de salida y ocultas. Una capa de entrada, también denominada sensorial, está compuesta por neuronas que reciben datos o señales procedentes del entorno. Una capa de salida se compone de neuronas que proporcionan la respuesta de la red neuronal. Las capas ocultas [17] son internas a la red y no tienen contacto directo con el entorno exterior.

4.2. Red neuronal multicapa

Una de las primeras redes que aparecieron a lo largo de la historia fue el perceptrón, la unidad básica desde donde nacería y se potenciarían las redes neuronales artificiales. El perceptrón es la forma más simple de una red neuronal usada para la clasificación de un tipo especial de patrones, los linealmente separables (es decir, patrones que se encuentran a ambos lados de un hiperplano). Básicamente, consiste en una neurona con pesos sinápticos y umbral ajustable.

La red neuronal multicapa surge como respuesta a las limitaciones que poseía un perceptrón simple, ya que con este solo se podían discriminar patrones que pudieran ser separados por un hiperplano - una recta. La forma de solventar estas limitaciones

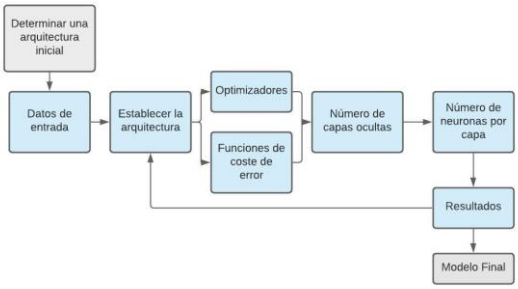

Fig. 1. Diagrama del proceso seguido para el desarrollo de la red neuronal.

fue a través de la inclusión de capas ocultas, permitiendo solucionar problemas que hasta ese momento no eran linealmente separables.

El perceptrón multicapa o MLP (Multi-Layer Perceptron, en inglés) se suele entrenar por medio de un algoritmo de retropropagación de errores o BP (Back Propagation, en inglés) de ahí que dicha arquitectura se conozca también bajo el nombre de red de retropropagación.

La popularidad de la arquitectura MLP se debe al hecho de que un MLP con una única capa oculta puede aproximar cualquier función continua en un intervalo hasta el nivel deseado, y que proporciona una base sólida al campo de las redes neuronales.

Los modelos con "profundidad" (decenas o cientos de capas) como las redes neuronales profundas han demostrado que pueden dar solución a un gran número de tareas, sin embargo, debido a su alta complejidad y altos requerimientos computacionales pueden ser demasiado difíciles de entrenar.

5. Desarrollo de la estructura de la red neuronal para ciberseguridad en el CAN bus

Definir la estructura de una red neuronal puede ser una tarea laboriosa debido a la falta de un protocolo o alguna estrategia establecida, la cual especifique que estructura se debe utilizar para cada problema bajo consideración (como el número de capas ocultas y el número de neuronas en cada capa).

En muchas ocasiones, esto se puede deducir mediante experimentos de prueba y error o incluso a través de la experiencia que pudiera tener el programador.

Para validar la estructura de la red neuronal que se utilizó en este trabajo, se realizaron una serie de experimentos, en los cuales se modificaron algunas características propias de la red neuronal**.**

En la Figura 1 se muestran las etapas seguidas para el desarrollo de la red neuronal propuesta en este trabajo. El proceso de validación de la red neuronal que se siguió consiste en realizar experimentos con el fin de obtener una arquitectura adecuada que proporcione buenos resultados. En este caso se modificó el número de capas, el número de neuronas, las funciones de coste de error y los optimizadores que intervienen en la arquitectura de una red neuronal**.**

Tabla 1. Número de mensajes que contiene los conjuntos de datos.

Conjunto de datos	No. de mensajes
Libre de ataques	2,369,868
Ataque tipo DoS	656,579
Ataque tipo Fuzzy	591,990
Ataque tipo Impersonation	995,472

En primera instancia, se determinó una arquitectura inicial con una sola capa oculta y sus correspondientes capas de entrada y de salida. Posteriormente se establecieron los conjuntos de datos que se utilizarían para alimentar a la red neuronal, así como la estructura que tendrían esto.

Una vez formada la red neuronal inicial ya con los datos de entrada establecidos, se comenzó a experimentar con las funciones de coste de error, así como con los optimizadores que forman parte de una red neuronal. En la literatura se han documentado muchas funciones, sin embargo, solo se eligieron las que son más comunes.

A continuación, se realizaron una serie de experimentos para determinar el número de capas ocultas y la cantidad de neuronas que tendría cada capa. En este paso se debe tomar en cuenta que entre más capas ocultas tenga la red neuronal mayor será su complejidad y como consecuencia se necesitarán más recursos computacionales.

Al comparar los resultados obtenidos con los casos anteriores, se determinará si hubo alguna mejora, y en caso de obtener mejores resultados, el proceso se hará de forma iterada hasta que los valores comiencen a converger a un valor. Cada una de las etapas que se siguieron proporcionó una arquitectura lo más sencilla posible con la cual se puede dar solución al problema planteado en este trabajo.

5.1. Conjunto de datos

Los datos utilizados durante este trabajo fueron recolectados por el laboratorio de investigación "Hacking and Countermeasure Research Lab (HCRL)" [13] y se encuentran disponibles para propósitos de investigación. Estos datos corresponden a sesiones de conducción de un automóvil realizados durante aproximadamente 40 minutos.

Durante estas pruebas se obtuvieron datos correspondientes a un tráfico en el CAN bus cuando no hay un ataque presente y cuando existen tres tipos de ataques: 1) DoS (acrónimo del inglés Denail of Service), 2) Fuzzy y 3) Impersonation (suplantación de identidad). La Tabla 1 muestra la cantidad de mensajes que se tiene por cada conjunto de datos.

Un DoS [15] es un ataque en el que se envía una gran cantidad de paquetes hacia una única ubicación. Esto generalmente conduce a una lentitud extrema o incluso a una parada completa del sistema. En un vehículo, un atacante podría inundar un componente importante. Por ejemplo, en el sistema de control electrónico del acelerador (ETC, del inglés Electronic Throttle Control) podría causar un mal funcionamiento y provocar un acelerador atascado o inoperativo.

Fig. 2. Estructura de un mensaje CAN.

ID_1 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_2 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_3 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_4 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_5 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_6 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_7 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_8 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_9 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8

Fig. 3. Estructura de las tramas que se presentan a la red neuronal.

En un ataque tipo Impersonation [14], el atacante busca engañar al sistema haciéndose pasar por una entidad fiable. Cuando se da este tipo de ataque en un vehículo se busca tener acceso a un sistema específico, por ejemplo, el sistema GPS. Este tipo de ataque permite que un atacante falsifique su posición geográfica haciendo que otros crean que el vehículo está en otra posición.

En un ataque tipo Fuzzy [15], el atacante envía mensajes engañando a los sistemas de control (en cualquier orden o aleatorio) utilizando datos de forma constante. Como resultado de ello, todos los nodos (ECU) de la red reciben muchos mensajes funcionales y se puede provocar un mal funcionamiento de la red. La Figura 2 muestra la estructura de un mensaje CAN, donde se pueden distinguir algunos campos importantes como el inicio de trama (SOF, del inglés Start Of Frame), un identificador ID del mensaje, un bit de "Solicitud de transmisión remota" (RTR) y un área llamada "campo de datos" (Data Field, en inglés).

Para el proceso de entrenamiento de la red neuronal, los datos fueron divididos en dos conjuntos: conjunto de entrenamiento y conjunto de prueba. La distribución que se tomó corresponde a un 80% para el conjunto de entrenamiento y un 20% para el conjunto de prueba. Para establecer los datos que se usaron en la capa de entrada de la RN, se formó una trama que consiste en nueve mensajes consecutivos.

Sin embargo, no todos los campos de un mensaje en el CAN bus son considerados, en este caso cada mensaje está conformado por el Arbitration ID (referido simplemente como ID) y el campo de datos (Data Field). El identificador ID es un elemento importante dentro de la red, ya que, a través de este las ECU son capaces de identificar solo aquellos mensajes que les pudiera servir. La Figura 3 muestra la estructura de la trama que se presenta a la entrada de la red neuronal.

5.2. Pruebas de validación de la red neuronal

Para determinar la arquitectura de la red neuronal que mejores resultados proporcione, se realizaron una serie de experimentos cuyo fin fue determinar las

Tabla 2. Resultados obtenidos durante las pruebas para una red neuronal con una sola capa oculta.

Optimizador	No. de épocas	No. de capas	No. de neuronas (capa oculta)	Exactitud (Entrenamiento)	Exactitud (Validación)
SGD	200	1	40	0.6515	0.6507
Adam	200	1	40	0.9099	0.9019
Adagrad	200	1	40	0.7021	0.7109

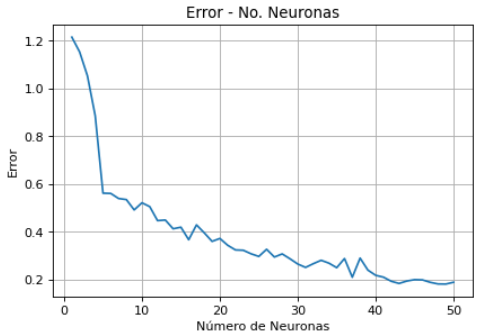

Fig. 4. Gráfica de error-número de neuronas en la capa oculta de la red neuronal propuesta.

funciones de coste de error, optimizadores, el número de capas y el número de neuronas, del modelo de red neuronal final.

De acuerdo con lo anterior, se armó una red neuronal tradicional con una capa de entrada (compuesta por 81 neuronas), una capa intermedia u oculta (con *n* neuronas), y una capa de salida (compuesta por 4 neuronas). Uno de los objetivos de estas pruebas consistió en identificar el número de capas ocultas que tendrá la red neuronal, así como el número de neuronas que tendrá cada una de ellas.

En primera instancia, se comenzó a realizar pruebas con los optimizadores y las funciones de coste de error. Los cuales se muestran a continuación.

Optimizadores:

– Adam

– Adagrad

– SGD

Funciones de Coste de error:

– Binary cross entropy

– Sparse categorical cross entropy

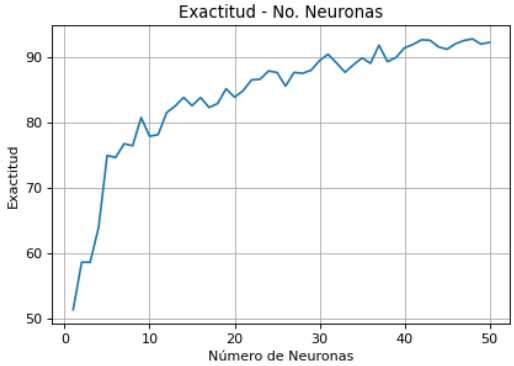

Fig. 5. Gráfica de exactitud-número de neuronas en la capa oculta de la red neuronal propuesta.

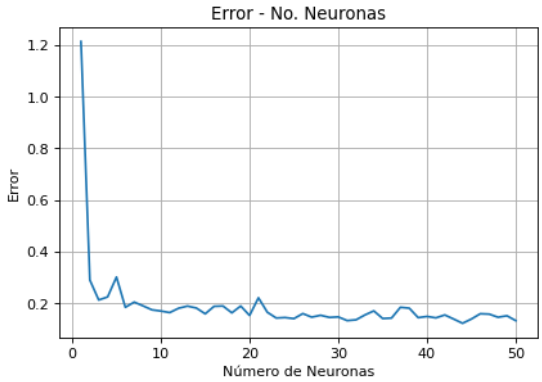

Fig. 6. Grafica de error-número de neuronas para la red neuronal con dos capas ocultas.

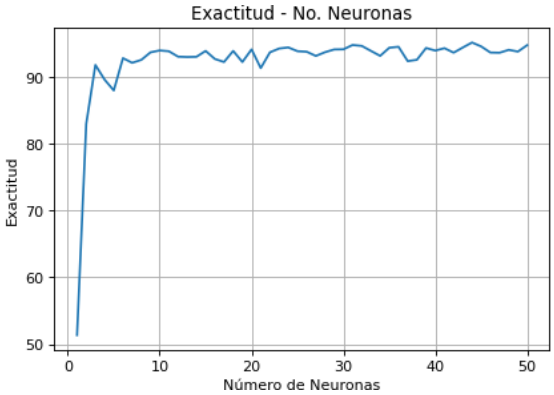

Fig. 7. Grafica de exactitud-número de neuronas para la red neuronal con dos capas ocultas.

Para la selección de la función de coste de error se comenzó con pruebas en las cuales se utilizó la función “Binary cross entropy” para realizar clasificación con solo 2 clases. Posteriormente se añadieron más clases para obtener una arquitectura multi-clase; es decir, datos que pertenecen tanto a un estado normal como a alguno de los tres ataques

Tabla 3. Resultados obtenidos durante las pruebas para una red neuronal con 2 capas ocultas.

Optimizador	**No. de épocas**	**No. de capas**	**No. de neuronas (1ª, 2ª capa)**	**Exactitud (Entrenamiento)**	**Exactitud (Validación)**
SGD	200	2	40,30	0.6036	0.6003
Adam	200	2	40,30	0.9257	0.9247
Adagrad	200	2	40,30	0.7165	0.7065

Tabla 4. Resultados obtenidos con diferente número de capas ocultas.

Capas Ocultas	No. de épocas	No. de neuronas (1ª, 2ª, …, 5ª capa)	Exactitud (Entrenamiento)	Exactitud (Validación)
1	200	40	0.9099	0.9019
2	200	40,30	0.9257	0.9247
3	200	40,30,28	0.9514	0.9379
4	200	40,30,28,26	0.9532	0.9404
5	200	40,30,28,26,24	0.9505	0.9315

considerados. Para este último, se utilizó la función de coste "Sparse categorical cross entropy". De forma simultánea, se experimentó con los optimizadores para determinar cuál de ellos se comportaba mejor.

Cada uno de los experimentos realizados arrojó resultados diferentes, en los cuales se pudo identificar las funciones que mejor se adaptaban a la arquitectura de la red neuronal, siendo en este caso el optimizador "Adam" y la función de coste "Sparse categorical cross entropy". En la Tabla 2 se muestran los resultados obtenidos durante esta prueba.

El número de neurona que tendría la capa oculta se determinó a través de experimentos en los cuales se entrenó la red neuronal desde una sola neurona hasta un total de 50. En la Figura 4 se presenta la gráfica correspondiente a la evolución del error al ir aumentando el número de neuronas en la capa intermedia, y en la Figura 5 se presenta la evolución del valor de exactitud obtenido al ir aumentando el número de neuronas en la capa intermedia.

En ambas figuras se puede observar que el error y la exactitud convergen aproximadamente después de tener 30 neuronas en la capa intermedia. En consecuencia, se optó por utilizar un total de 30 neuronas para estas primeras pruebas.

El siguiente paso fue realizar experimentos en el cual se modificó el número de capas que tendría la red neuronal. Los experimentos realizados fueron similares al que se hizo con una sola capa oculta, es decir, el número de neuronas en la segunda capa oculta se fue incrementando de una en una hasta encontrar convergencia en error y exactitud. Los resultados obtenidos durante estas pruebas se observan en las Figuras 6 y 7. En dichas figuras se puede apreciar la evolución del error y de la exactitud al ir incrementando el número de neuronas.

Tabla 5. Resultados obtenidos para una red neuronal bi-clase.

Caso	No. de épocas	No. de capas ocultas	No. de neuronas (1ª, 2ª y 3ª capa)	Exactitud (Entrenamiento)	Exactitud (Validación)
1	200	3	40,30,28	0.9999	0.9995
2	200	3	40,30,28	0.9989	0.9951
3	200	3	40,30,28	0.9623	0.9500

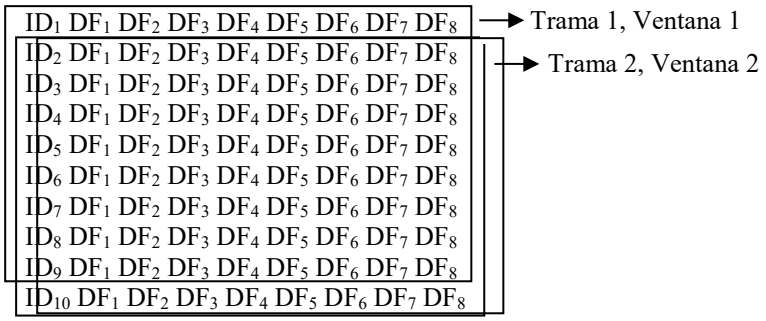

Fig. 8. Esboce de las tramas que se forman al tomar en cuenta las ventanas deslizantes.

De acuerdo con los resultados mostrados en las gráficas se puede observar que se requieren un total de 20 neuronas para la segunda capa intermedia. En la Tabla 3, se muestran los resultados obtenidos con una red neuronal que tiene dos capas ocultas. De estas pruebas, se observa que hay una mejora en el valor de exactitud obtenido en la red neuronal. Se obtiene un 92 % de exactitud comparado con el valor obtenido para una red neuronal con una sola capa oculta (90 %).

El siguiente paso fue realizar pruebas en las cuales se aumentaron el número de capas ocultas y posteriormente el número de neuronas para cada caso.

El proceso para determinar el número de neuronas que tendría la tercera, cuarta y quinta capa se realizó de forma similar a la descrita en la red con una sola capa. Los resultados obtenidos en estas pruebas se muestran en la Tabla 4.

De la tabla anterior se puede concluir que el número de capas ocultas adecuado para resolver el problema en cuestión es de 3, debido a que el valor de exactitud se mantiene o disminuye después de este número (al agregar más capas intermedias). De esta forma es posible descartar una arquitectura con un número mayor de capas ocultas con el fin de tener una arquitectura lo más ligera posible.

5.3. Pruebas con redes neuronales bi-clase

Las siguientes pruebas consistieron en entrenar la red neuronal de tal forma que solo se obtuviera un clasificador bi-clase. Es decir, una red que clasifique datos libres de ataque o datos de algún tipo de ataque. Los clasificadores bi-clase considerados son:

1. Datos libre de ataque – Datos de ataque DoS
2. Datos libre de ataque – Datos de ataque Fuzzy

Tabla 6. Resultados obtenidos para una red neuronal con ventanas deslizantes.

Caso	No. de épocas	No. de capas ocultas	No. de neuronas (1ª capa, 2ª capa, 3ª capa)	Exactitud (Entrenamiento)	Exactitud (Validación)
4 clases	200	3	30,20,18	0.9602	0.9540
3 clases	200	3	30,20,18	0.9971	0.9972
DOS – Free	200	3	30,20,18	0.9999	0.9998
Fuzzy – Free	200	3	30,20,18	0.9983	0.9976
Impersonation – Free	200	3	30,20,18	0.9731	0.9604

Tabla 7. Resultados obtenidos en otros trabajos relacionados.

Articulo	Exactitud	Precisión	Estructura
[2] DCNN	99 % (DOS,Spoofing, Fuzzy)	99 %	3 filtros (varias capas convolucionales)
[3] DNN	98 % (Inyección de mensajes)	98 %	11 capas ocultas
[4] GAN	98 % (DOS,Spoofing, Fuzzy)	98 %	2 RN (convolucional, profunda)
Propuesta	99 % (DOS,Impersonation, Fuzzy)	99 %	3 capas ocultas

3. Datos libre de ataque – Datos de ataque Impersonation

Los resultados que se obtuvieron se observan en la Tabla 5. En dicha tabla se puede observar que hay una mejora en el valor de exactitud para los casos en que hay una red neuronal bi-clase.

5.4. Pruebas con ventanas deslizantes

Finalmente, se realizaron pruebas en las cuales los datos que se usan en la capa de entrada son tratados en un formato de ventana deslizante. En este caso, cada trama tiene la característica que el mansaje más antiguo se descarta de la trama y el nuevo mensaje que llega al CAN bus se añade a la misma trama.

En este caso, a diferencia de las pruebas anteriores donde se presenta una trama de 9 mensajes en la que se tiene que esperar para tener la cantidad necesaria que formará la trama, la red neuronal que procesa el patrón de entrada en un formato de ventana deslizante tiene la capacidad de formar una trama vez que llega un nuevo mensaje.

Este formato puede ser utilizado para llevar a cabo experimentos en tiempo real ya que no hay necesidad de esperar a tener los 9 mensajes, es decir, cuando un mensaje nuevo llega al CAN bus la trama se actualizará con este nuevo mensaje y el más antiguo

se descarta. Además, se puede observar una mejora en el valor de exactitud al utilizar los datos de entrada como ventanas deslizantes. En la Figura 8 se presenta un esboce de las tramas que se forman al tomar en cuenta las ventanas deslizantes. Tomando en cuenta la estructura de los datos en un formato de ventana deslizante, se realizaron experimentos en los cuales se utilizó dicho formato. Los resultados obtenidos durante las pruebas se muestran en Tabla 6.

En la misma Tabla se puede observar que una red neuronal con ventanas deslizantes proporciona mejores valores de exactitud comparado con los experimentos anteriores. Además, las redes neuronales bi-clase tienen mejor comportamiento que cuando se tiene una red multiclase. Una red neuronal que posee datos de ataque tipo Impersonation otorga valores de exactitud más bajos. Por tal motivo, se puede decir que los datos de este tipo de ataque poseen una complejidad mayor y en consecuencia se puede proponer una arquitectura diferente para este caso.

6. Conclusiones y trabajo a futuro

Desarrollar una red neuronal puede ser una tarea difícil y los resultados pueden variar en cada proyecto. En este trabajo se ha propuesto un IDS basado en técnicas de aprendizaje automático para detectar datos anormales en el CAN bus de un automóvil, diferenciando de cuando se observa un comportamiento normal.

El MLP propuesto puede discriminar datos normales y de 3 tipos de ataques. Esta red neuronal tiene un buen desempeño para detectar datos libres de ataque, DoS, Fuzzy e Impersonation. Se puede observar que es posible obtener resultados adecuados para los casos en que se utiliza una red neuronal bi-clase y una red neuronal con ventanas deslizante.

Los mejores resultados obtenidos se presentan cuando se utiliza este tipo de clasificador y una ventana deslizante en la configuración de los patrones de entrada que se le presentan a la red neuronal, con una estructura de 3 capas ocultas.

Las exactitudes obtenidas en este caso son del 99 % para una red neuronal bi-clase para los casos de ataque DoS y Spoofing, 97 % para un ataque tipo Fuzzy y del 98 % para una red neuronal multiclase.

También podemos observar que se obtienen buenos resultados de exactitud sin la necesidad de contar con una red neuronal compleja como las redes neuronales convolucionales o las redes profundas. La Tabla 7 muestra los valores de exactitud que se obtuvieron en los trabajos relacionados, así como los resultados obtenidos en este trabajo. El trabajo que faltaría realizar consiste en realizar pruebas en tiempo real para un caso simulado e incluso a futuro realizarlo en un vehículo real.

Agradecimientos. Los autores agradecen el apoyo de CONACYT, y al IPN bajo los proyectos SIP-1999 y SIP-20210039

Referencias

1. Baresi, L., Colazzo, S., Mainetti, L., Morasca, S.: W2000: A modelling notation for complex web applications. Mendes, E., Mosley, N. (Eds): Web Engineering, Springer, pp. 335–364 (2006) doi: 10.1007/3-540-28218-1_11

2. Miller, C., Valasek, C.: A survey of remote automotive attack surfaces. black hat USA, no. 94 (2014)
3. Song, H., M., Woo, J., Kim, H. K.: In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications, vol. 21, pp. 100–198 (2020) doi: 10.1016/j.vehcom.2019.100198
4. Kang, M. J., Kang, J. W.: Intrusion detection system using deep neural network for in-vehicle network security. PloS one, vol. 11, no. 6, (2016) doi: 10.1371/journal. pone.0155781
5. Seo, E., Song, H., M., Kim, H., K.: Gids Gan based intrusion detection system for in-vehicle network. In: Proceedings of 16th Annual Conference on Privacy, Security and Trust IEEE, pp. 1–6 (2018) doi: 10.1109/PST.2018.8514157
6. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: 2016 IEEE International Conference on Data Science and Advanced Analytics, pp. 130–139 IEEE (2016) doi: 10.1109/DSAA. 2016.20
7. Ilakkiya, B., Vanitha, V.: A survey on engine control unit. vol. 2, no. 3, pp. 21–24 (2016)
8. Rizvi, S., Willet, J., Perino, D., Marasco, S., Condo, C.: A threat to vehicular cyber security and the urgency for correction. Procedia Computer Science, vol. 114, pp. 100–105 (2017) doi: 10.1016/j.procs.2017.09.021
9. Ogle, A.: Automotive controller area network systems: Driver convenience or cyber security threats? Utica College ProQuest Dissertations Publishing (2017)
10. Markovitz, M., Wool, A.: Field classification, modeling and anomaly detection in unknown CAN bus networks. Vehicular Communications, vol. 9, pp. 43–52. (2017) doi: 10.1016/ j.vehcom.2017.02.005
11. Li, H., Zhao, L., Juliato, M., Ahmed, S., Sastry, M. R., Yang, L. L.: Poster intrusion detection system for in-vehicle networks using sensor correlation and inte-gration. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 2531–2533 (2017) doi: 10.1145/3133956.3138843
12. Miller, C., Valasek, C.: A survey of remote automotive attack surfaces. Black hat USA, (2014)
13. Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat USA (2015)
14. Hacking and countermeasure research lab. (https://ocslab.hksecurity.net/Datasets)
15. Kosmanos, D., Pappas, A., Maglaras, L., Moschoyiannis, S., Aparicio-Navarro, F. J., Argyriou, A., Janicke, H.: A novel intrusion detection system against spoofing attacks in connected electric vehicles. Array, vol. 5, (2020) doi: 10.1016/j.array.2019.100013
16. Alshammari, A., Zohdy, M. A., Debnath, D., Corser, G.: Classification approach for intrusion detection in vehicle systems. Wireless Engineering and Technology, vol. 9, no. 4, pp. 79–94 (2018) doi: 10.4236/wet.2018.94007
17. Sanchez-Vela, L. G., Molano-Clemente, M. J., Fabela-Gallegos, M. J., Martinez-Madrid, M., Hernandez-Jimenez, J. R., Vazquez-Vega, D.: Revisión documental del protocolo can como herramienta de comunicación y aplicación en vehículos. Publicación Técnica, no. 474 (2016)
18. Matich, D. J.: Redes neuronales: Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional (2001)