CAPÍTULO 6: PRUEBAS Y RESULTADOS

En este capítulo se proporciona una descripción de las pruebas a las que fue sometido el sistema de detección de intrusos, y se evalúan estos valores comparativamente para demostrar que un IDS basado en RNN es superior a otras arquitecturas de diseño, específicamente las que involucran redes *feedforward* y redes Elman con base en sus medidas de velocidad, exactitud y porcentajes de correcta clasificación e identificación de ataques hechos por intrusos.

6.1 PRUEBAS

Idealmente, la salida del IDS que hemos diseñado al final de cada cadena de comandos basados en HTTP que se le presente debería indicar de qué tipo de ataque (si es que es un comando peligroso) se trata asignándole un valor de 1 a la neurona de salida correspondiente y fijando en cero todos los demás valores de las otras cuatro neuronas. No obstante, en la práctica la salida de cada cadena de prueba da como resultado valores decimales cuyo rango de valores oscila entre 0 y 1; en esos casos un vector de salida típico incluirá valores decimales en el intervalo $0 \leq x \leq 1$.

Este tipo de salidas no es útil para nosotros, que esperamos únicamente valores binarios, por lo que haremos uso de una función en MATLAB® denominada función de competencia [10]. El propósito de esta función es que, dado un vector de valores determinado, el valor más grande de entre ellos tendrá un valor de 1, mientras que todos los demás tendrán un valor de cero. De esta manera, la salida anterior sería transformada como en la Figura 6.1:

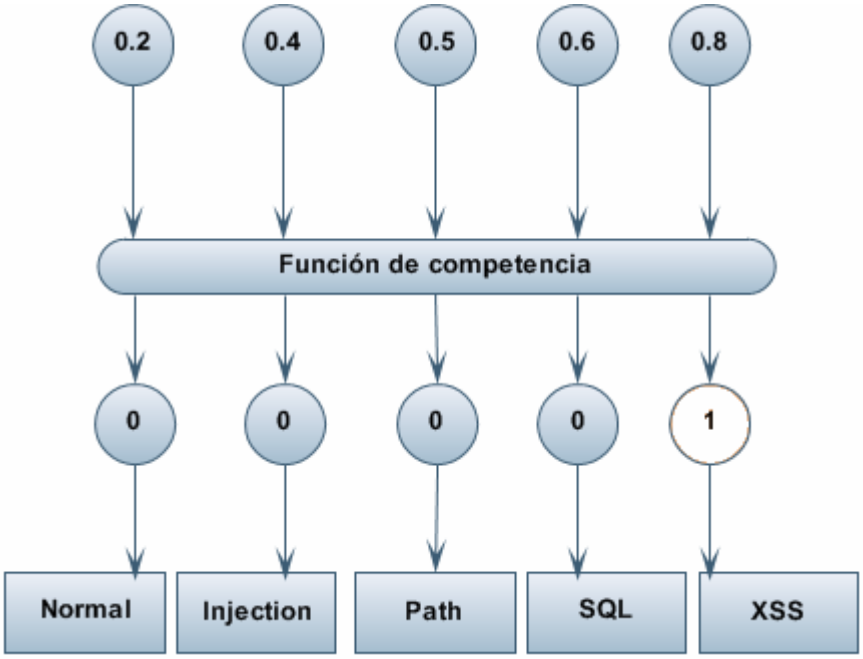

Figura 6.1. Salida después de la función de competencia

Para este vector de salida en particular, la red neuronal ha identificado un ataque tipo XSS y con base en este resultado puede tomarse algún tipo de acción para prevenir otros ataques similares. También es importante señalar el hecho de que, al usar vectores de tamaño fijo a la entrada de las tres redes (64 elementos), algunas cadenas con un número de elementos superior a 64 / 8 = 8 elementos quedaron divididas en dos o más cadenas equivalentes representando la misma cadena de comandos [21]. Teóricamente, esperamos el mismo resultado de la red para ambas cadenas, lo cual se cumplió en la mayoría de los casos. Sin embargo, algunas veces la red identificó correctamente una cadena como ataque pero erró al clasificarla en la categoría correspondiente. Por ejemplo, una cadena que representa un ataque SQL pudo ser clasificada como un ataque de inyección de código a pesar de no serlo y, aunque el IDS ha identificado correctamente la cadena como anormal, no así ha podido hacerlo como el tipo correcto de intrusión.

Para tomar en cuenta estos errores inevitables en el desempeño de las redes se definirán dos porcentajes diferentes para evaluar los resultados de las mismas: porcentaje de clasificación y porcentaje de identificación. El porcentaje de clasificación corresponderá al número de veces que la red ha determinado correctamente que una cadena de caracteres representa un ataque contra representar un comando normal. Para ello, basta con que un ataque no sea clasificado como Normal para que el IDS se desempeñe correctamente. El porcentaje de identificación, por otro lado, corresponderá al número de veces que cualquiera de las redes neuronales determinen correctamente a cuál de las cinco categorías pertenece un ataque. Esto es, que si una cadena representa la categoría *Path*, se le identifique como tal.

La relación entre ambos criterios se ilustra en la Figura 6.2:

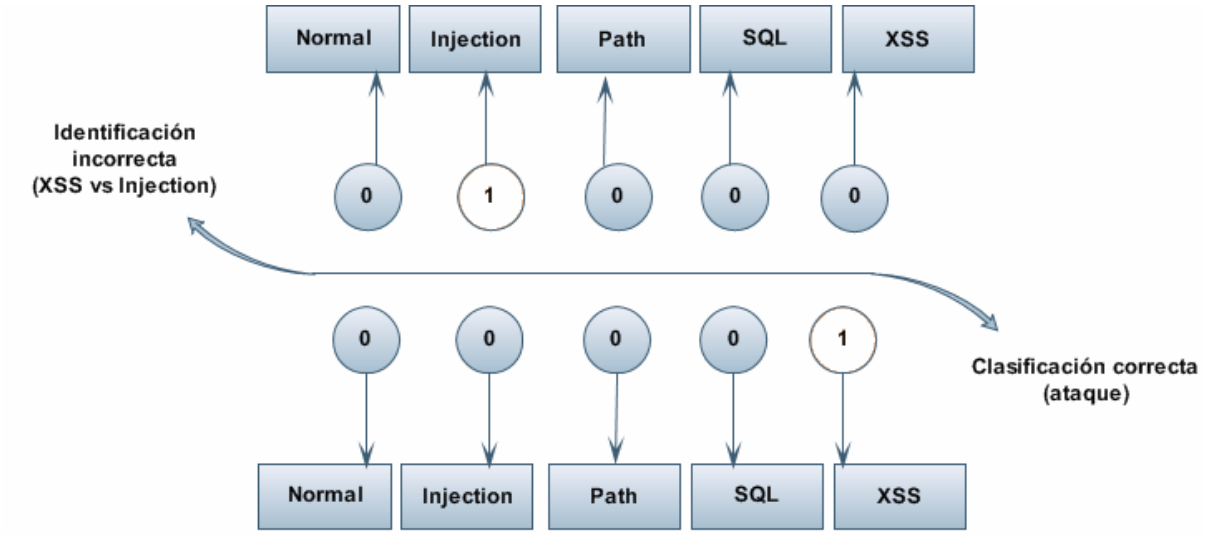

Figura 6.2. Clasificación contra identificación

Puede verse que esperamos que el porcentaje de clasificación sea mayor al porcentaje de identificación, ya que basta con que la red determine si un vector es ataque o no para obtener un desempeño correcto y es más difícil que la red acierte a

identificar cuál de las cinco categorías es a la que pertenece un vector de entrada determinado en formato binario.

6.2 RESULTADOS

Conforme se probaron las tres diferentes redes neuronales se calcularon sus porcentajes de clasificación e identificación para hacer una tabla comparativa en donde puedan apreciarse las ventajas y desventajas de cada una. Se incluyó también información referente a la velocidad de procesamiento, el número de *epochs* transcurridos antes de llegar al valor máximo de error permitido, y también datos referentes al índice de *false positive* y *false negative* del IDS.

Como su nombre lo indica, un *false positive* corresponde al porcentaje de veces que el sistema identifica una acción válida como un ataque, resultando en una respuesta inapropiada del sistema de control conectado al IDS [11].

$$false\ positive = \frac{num.\ de\ falsos\ positivos}{num.\ de\ instancias\ negativas} \tag{16}$$

Un *false negative* representa la situación en la que el sistema no logra identificar un ataque como tal y se permite que un comando originado por un intruso ingrese a la red de computadoras sin haber sido detectado [11, 21].

$$false\ negative = \frac{num.\ de\ falsos\ negativos}{num.\ de\ instancias\ positivas} \tag{17}$$

Basados en los resultados anteriores de entrenamiento y la velocidad de convergencia para los tres tipos de redes, esperamos que la RNN tenga el mejor desempeño y los porcentajes de *false positives* y *false negatives* más bajos. Esto es debido a la arquitectura de la red, así como al algoritmo de entrenamiento utilizado

(RTRL), el cual es superior al algoritmo de *Backpropagation* en aplicaciones tales como el desempeño *on-line* de un sistema de detección de intrusos [11].

Los resultados finales se muestran a continuación.

Tabla 6.1. Tabla comparativa de resultados finales

Tipo de red	*Feedforward* 64x15x15x5	Elman 64x30x30x5	RNN 64x30x5
Error	0.2572	0.2161	0.1735
Epochs	239	125	75
Clasificación	0.8788	0.8961	0.9409
Identificación	0.8355	0.8412	0.8722
O(n)	n^2	n^2	n^4
False negative	0.95%	0.92%	0.87%
False positive	4.94%	4.90%	4.73%

Puede apreciarse con claridad que la red neuronal que mejor desempeño tuvo en todas las categorías excepto una fue la RNN completamente conectada. Su error global fue menor que el de las demás, y sus índices de clasificación, identificación y falsos positivos y negativos también fueron superiores. Esto concuerda con nuestras expectativas iniciales. En la única categoría en que la RNN presenta una desventaja es precisamente en la de costo de procesamiento. Puede verse que n^4, donde n es el número de neuronas en toda la red, es un costo más bien elevado que se vuelve muy significativo cuando el número de neuronas se incrementa en ciertas aplicaciones. Por

estas razones, en el problema específico del IDS se seleccionó utilizar una red de este tipo con relativamente pocas neuronas y pocas capas ocultas para minimizar este efecto y así poder tener los resultados positivos de esta arquitectura sin tener que preocuparnos de que el procesador encargado de administrar la operación del programa tenga que realizar cálculos excesivos que contrarresten las características buenas del sistema.

6.3. RESUMEN

En este capítulo se ha demostrado que las redes neuronales recurrentes son superiores a los otros dos tipos de redes utilizadas con fines de comparación. Todas las redes mostraron cierto número de errores que se tradujeron en porcentajes de Clasificación, Identificación y falsos positivos y negativos. El cuadro comparativo incluido hizo patente la superioridad de la RNN sobre las demás.

Los datos que arrojaron las redes a la salida inicialmente no fueron los que necesitábamos, ya que nosotros requerimos valores binarios únicamente. Por esto incluimos una función de competencia a la salida del sistema cuya función era asignar un valor de uno a la salida de la neurona con valor mayor y cero a todas las demás, facilitando el procesamiento de los resultados.

A pesar de que una red clasifique correctamente un vector de entrada como ataque o dato normal, es más difícil que se le coloque a dicho vector en la categoría correcta de entre las cinco posibles. Además, existen ocasiones en que la red neuronal ignora un ataque cuando en realidad lo hubo, y en ese caso hablamos de un falso negativo, lo cual trae consecuencias graves al sistema de detección de intrusos como un todo. Afortunadamente, en la práctica estos porcentajes fueron muy pequeños y no hay razón para considerarlos un gran problema.

Desde el punto de vista de los valores finales que alcanzaron las redes tras ser entrenadas y posteriormente probadas con datos totalmente nuevos, es claro que la RNN completamente conectada las aventaja en todos los parámetros excepto uno: el costo de procesamiento por neurona. Una red recurrente con un alto número de neuronas se convierte rápidamente en algo muy demandante para un procesador dado porque el tiempo se incrementa a la cuarta potencia con respecto al número de elementos del sistema.

En cuanto a la viabilidad de desarrollar una red de este tipo, se ha demostrado que los algoritmos de entrenamiento son tales que un IDS basado en ellos no requerirá recursos excesivos del procesador para ser exitoso.