UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA

TESIS DE GRADO

"DETECCIÓN DE FRAUDE EN TELEFONÍA CELULAR USANDO REDES NEURONALES"

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

AUTOR	:	Univ. María Elena Sea Ali
TUTOR	:	Lic. Efraín Silva Sánchez
REVISOR	:	Mg.Sc Carlos Mullisaca Choque

LA PAZ – BOLIVIA
2011

Dedicatoria

A mis queridos padres Dionicio y Fortunata quienes me apoyaron, me brindaron su cariño y me dieron aliento para seguir adelante.

A mis hermanos por brindarme su apoyo en todo momento. Y a la persona tan especial para mí que me enseño a seguir adelante a pesar de los tropiezos que se tiene en la vida, al Ing. Guillermo Callisaya Cutipa.

Agradecimientos

A Dios por darme la vida y por haberme guiado y permitido que lograra uno más de mis objetivos.

A mi docente Tutor Lic. Efraín Silva Sánchez, por haberme brindado la colaboración con toda su capacidad y conocimiento en la realización de la presente Tesis de Grado.

A mi docente Revisor Lic. Carlos Mullisaca Mg.Sc., por el asesoramiento, paciencia y revisión que demostró toda su capacidad y conocimiento para la elaboración de la Tesis de Grado.

Al Ing. Guillermo Callisaya Cutipa por haberme apoyado con sus conocimientos y brindarme sus valiosos consejos.

Por último quiero agradecer a todos los docentes de la carrera de Informática y también a la “Universidad Mayor de San Andrés” por los años que me cobijo en sus aulas, para adquirir conocimiento y tener una formación profesional.

e-mail: mari_sea210902@hotmail.com

RESUMEN

En este trabajo se aborda el problema de la detección de cambios de consumo de usuarios de telefonía celular fuera de lo normal, la correspondiente construcción de estructuras de datos que representen el comportamiento reciente e histórico de cada uno de los usuarios, teniendo en cuenta la información que contiene una llamada. Si bien existen diferentes formas de detectar fraude, todas ellas trabajan con picos de consumo o reglas fijas, que no siempre indican comportamiento fuera de lo normal. La solución que se presenta utiliza la tecnología de redes neuronales no supervisadas, en particular las redes SOM.
Palabras clave: Detección de Fraude, Redes Neuronales.

ABSTRACT

This work deals with the problem of detection of changes in cell phones' usage for users out of the normal behavior, the developing of structures of data that represent the recent and historic behavior of each user, taking into account the information that resides in a call and the complexity of the development of a function with so many imput variables where the parameterization is not always known. Even though several fraud detection tools have been developed, all of them evaluate high usage variables or fixed rules, that not always indicate non normal behaviour. The solution that is presented uses the technology of non supervised neural networks, in particular the SOM networks.
Kewords: Fraud Detection, Neural Networks.

INDICE

CAPITULO I 9
1.1. INTRODUCCION 9
1.2. ANTECEDENTES 10
1.3. PLANTEAMIENTO DEL PROBLEMA 15
1.3.1. FORMULACION DEL PROBLEMA 16
1.4. HIPOTESIS 16
1.5. OBJETIVOS 16
1.5.1. OBJETIVO GENERAL 16
1.5.2. OBJETIVOS ESPECIFICOS 16
1.6. JUSTIFICACION 17
1.6.1. JUSTIFICACION CIENTIFICA 17
1.6.2. JUSTIFICACION SOCIAL 17
1.6.3. JUSTIFICACION ECONOMICA 17
1.7. METODOLOGIA 17
1.8. ALCANCES Y APORTES 18
1.8.1. ALCANCES 18
1.8.2. APORTES 18
CAPITULO II 20
MARCO TEORICO 20
2.1.1. CLASIFICACION DE TIPOS DE FRAUDE 20
2.1.2. FRAUDE CONTRACTUAL 20
2.1.3. FRAUDE POR VIOLACION DE SEGURIDAD 21
2.1.4. FRAUDE TECNICO 21
2.1.5. FRAUDE DE PROCEDIMIENTO 22
2.2. DETECCION DE FRAUDE EN TELEFONIA CELULAR 23
2.2.1. ENFOQUE POR ENSEÑANZA 25
2.2.2. ENFOQUE POR APRENDIZAJE 25
2.3. REDES NEURONALES 26
2.4. INTRODUCCION A LAS REDES NEURONALES ARTIFICIALES 27
2.4.1. ELEMENTOS DE UNA RED NEURONAL ARTIFICIAL 28
2.4.2. TOPOLOGIA DE LAS REDES NERONALES ARTIFICIALES 29
2.4.3. MECANISMO DE APRENDIZAJE 31
2.4.3.1. REDES CON APRENDIZAJE SUPERVISADO 32
2.4.3.2. REDES CON APRENDIZAJE NO SUPERVISADO 33
2.4.4. CLASIFICACION DE REDES NEURONAES 34
2.5. RED NEURONAL SUPERVISADA – EL PERCEPTRON 35
2.6. SELF ORGANIZING MAPS (SOM) 36
2.6.1. ALGORITMO DEL SOM 37
2.6.1.1. PRE-PROCESAMIENTO DE LOS DATOS 39
2.6.1.2. INICIALIZACION 40
2.6.1.4. VISUALIZACION 45
2.6.1.5. VALIDACION 46
2.6.2. APLICACIONES 46
2.6.3. PREDICCION DE CAMPOS ESTOCASTICOS GENERADOS POR REDES 46

2.7. ANALISIS DE LA INFORMACION PARA LA DETECCION DE FRAUDE 47
2.8. ENFOQUES PARA LA DETECCION DE FRAUDE 49
2.8.1. ENFOQUE BASADO EN REGLAS 49
2.8.1.1. NATURALEZA ADAPTATIVA DE LA SOLUCION 49
2.8.1.2. MODELO DE LA SOLUCION POR REGLAS 50
2.8.1.3. LIMITACIONES DEL ENFOQUE POR REGLAS 52
2.8.2. ENFOQUE BASADO EN REDES NEURONALES 52
2.8.2.1. MODELO UTILIZANDO REDES NEURONALES SUPERVISADAS 53
2.8.2.2. LIMITACIONES DEL ENFOQUE BASADO EN REDES 55
2.8.2.3. MODELO UTILIZANDO REDES NEURONALES NO SUPERVISADAS 55
2.9. MARCO LEGAL 56
CAPITULO III 60
MARCO APLICATIVO 60
3.1. INTRODUCCION 60
3.2. ANALISIS DE LA INFORMACION PARA DETECCION DE FRAUDE 61
3.3. SOLUCION A LA CONSTRUCCION DE “PERFILES DE USUARIO” 62
3.4. SOLUCION A LA DETECCION DE CAMBIOS DE COMPORTAMIENTO 65
3.5. SOLUCION A LAS CUESTIONES DE PERFORMANCE 66
3.6. METOLOGIA UTILIZADA 66
3.6.1. EXPERIMENTOS DE GENERACION DE PATRONES 66
3.6.2. EXPERIMENTOS DE CONSTRUCCION DE PERFILES Y DETECCION DE 67
3.7. PARAMETROS UTILIZADOS PARA LA GENERACION DE PATRONES 68
3.7.1. PARAMETROS INDEPENDIENTES 68
3.7.2. PARAMETROS DEPENDIENTES 68
3.8. PARAMETROS UTILIZADOS PARA LA CONSTRUCCION DE PERFILES 68
3.9. RESULTADOS 69
3.9.1. GENERACION DE PATRONES 69
3.9.2. CONSTRUCCION DE PERFILES Y DETECCION DE CAMBIOS DE 71
CAPITULO IV 77
CONCLUSIONES Y RECOMENDACIONES 77
4.1. CONCLUSIONES 77
ANEXOS 82

INDICE DE FIGURAS Y GRAFICOS

FIGURAS

FIGURA 2.1 Análisis Absoluto vs. Análisis Diferencial……23
FIGURA 2.2 Estados de una neurona……28
FIGURA 2.3 Topologías de redes neuronales……29
FIGURA 2.4 Un Perceptrón Multicapa……35
FIGURA 2.5 Estructuras de los mapas……37
FIGURA 2.6 Vecindario de una neurona……38
FIGURA 2.7 Funciones de vecindario……42
FIGURA 2.8 Tasas de aprendizaje……42
FIGURA 2.9 U-MATRIX……44
FIGURA 2.10 Enfoque basado en reglas……49
FIGURA 2.11 Enfoque basado en redes neuronales supervisadas……53

GRAFICOS

Gráfico 3.1: Patrones llamadas locales……69
Gráfico 3.2: Patrones llamadas nacionales……69
Gráfico 3.3: Patrones llamadas internacionales……70
Gráfico 3.4: Distribución de frecuencias CUP experiencia 1……71
Gráfico 3.5: Distribución de frecuencias UPH experiencia 1……72
Gráfico 3.6: Distribución de frecuencias CUP experiencia 2……73
Gráfico 3.7: Distribución de frecuencias UPH experiencia 2……73

CAPITULO I
INTRODUCCION

CAPITULO I

INTRODUCCION

1.1. INTRODUCCION

En los últimos años de exploración en inteligencia artificial, los investigadores se han intrigado por las redes neuronales. Como su nombre lo implica, una red neuronal artificial consiste en una red de neuronas artificiales interconectadas. El concepto se basa vagamente en cómo pensamos que funciona el cerebro de un animal. Un cerebro consiste en un sistema de células interconectadas, las cuales son, aparentemente, responsables de los pensamientos, la memoria y la conciencia. Las neuronas se conectan a muchas otras neuronas formando uniones llamadas sinapsis; las señales electroquímicas se propagan de una neurona a otra a través de estas sinapsis. Las neuronas demuestran plasticidad: una habilidad de cambiar su respuesta a los estímulos en el tiempo, o aprender; en una red neuronal artificial, se imitan estas habilidades por software [Hilera González & Martínez Hernando, 2000].

Las Redes Neuronales Artificiales son redes de elementos simples interconectadas masivamente en paralelo y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico [Kohonen, 1988]. La compleja operación de las redes neuronales es el resultado de abundantes lazos de realimentación junto con no linealidades de los elementos de proceso y cambios adaptativos de sus parámetros, que pueden definir incluso fenómenos dinámicos muy complicados [Hilera González & Martínez Hernando, 2000]. Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante [Hilera González & Martínez Hernando, 2000].

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero. Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables o el margen de error es menor o igual al que se ha definido como aceptable.

Un aspecto importante respecto al aprendizaje en las redes neuronales es el conocer cómo se modifican los valores de los pesos; es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red *aprenda* una nueva información. Estos criterios determinan lo que se conoce como la *regla de aprendizaje* de la red. De forma general, se suelen considerar dos tipos de reglas: las que responden a lo que habitualmente se conoce como aprendizaje supervisado, y las correspondientes a un aprendizaje no supervisado. La diferencia fundamental entre ambos tipos de aprendizaje está en la existencia o no de un agente externo (*supervisor*) que controle el proceso de aprendizaje de la red.

El presente trabajo se encuentra orientado a la construcción de un modelo neuronal, para la detección de fraude en telefonía celular, con el cual se busca colaborar a las telefonías celulares, aprovechando la gran ventaja que proporciona las Redes Neuronales.

1.2. ANTECEDENTES

Fraude se puede describir de una manera simple como "cualquier actividad por la cual un servicio es obtenido sin la intención de pagarlo". Gosset & Hyland, 1999].

Muchas veces las organizaciones calculan cuánto dinero pierden debido al fraude definiéndolo como "el dinero que se pierde en clientes/cuentas por los cuales no se recibe ningún pago" [Gosset & Hyland, 1999]. Sin embargo, para los fines de

detección, tal definición no es apropiada debido que el fraude solo sería detectado una vez que ha ocurrido. De hecho, especificar qué es el fraude puede ser muy difícil, debido a que la diferencia entre un comportamiento fraudulento y uno que no lo es puede ser muy pequeña; por lo tanto lo más prudente es clasificar al fraude en diferentes tipos y describir cada uno de ellos.

Las formas de realizar fraude están constantemente evolucionando y cambiando; esto se debe a que la tecnología en telecomunicaciones avanza y restringe cada vez más las posibilidades de cometer actos fraudulentos. Cuando las primeras redes móviles de comunicaciones analógicas fueron lanzadas al mercado, su debilidad principal residía en la seguridad, particularmente en la falta de encriptación de los datos en los canales de comunicación que permitía la clonación de teléfonos celulares (dos aparatos diferentes usando la misma cuenta). A medida que la tecnología evolucionó de analógica a digital, la naturaleza del fraude ha cambiado haciéndose más difícil la clonación, y llevando estas actividades hacia otros tipos de fraude; sin embargo, tampoco las redes digitales están libradas completamente del fraude de clonación.

La pérdida anual en la industria global de las telecomunicaciones debido al fraude se estima en decenas de millones de dólares [Taniguchi, Haft, Hollmen & Tresp, 1998]. Esto hace que la detección y prevención del fraude sea una actividad muy importante. En las siguientes secciones se presentan algunos conceptos que serán desarrollados en profundidad a lo largo de todo el trabajo.

La proliferación de este tipo de dispositivos, sumada al hecho de que son una pequeña computadora con debilidades para explotar, hace que los ciberdelincuentes enfoquen en ellos sus esfuerzos para cometer fraudes, lo que aumenta el número de amenazas para estas plataformas. De acuerdo al último informe de Virología móvil publicado Kaspersky Lab, el desarrollo de programas dañinos se centra en entornos multiplataforma, dada la ausencia de un software líder.

Kaspersky Lab publicó la tercera parte de su informe Virología móvil dedicado a los programas nocivos que amenazan los móviles y smartphones, cuya primera parte se publicó hace ya tres años. Durante este tipo, los expertos de Kaspersky

Lab han presenciado cambios importantes en el mundo de los dispositivos móviles. En el mercado de sistemas operativos para teléfonos móviles y smartphones se nota la ausencia de un líder claro. Con el anterior líder, Symbian, compiten con Windows Mobile, BlackBerry, la versión móvil de MacOS X y otras plataformas.

Esta situación es muy diferente a las condiciones del mercado de las computadoras de escritorio, donde domina Microsoft Windows. Debido a la ausencia de una plataforma líder, los creadores de programas nocivos para dispositivos móviles se han encontrado con la imposibilidad de realizar ataques masivos para la mayoría de teléfonos y smartphones. Esto ha llevado al desarrollo de tecnologías para amenazas móviles aplicables a multiplataformas.

La tecnología Java 2 Micro Edition, que asegura la funcionalidad del lenguaje Java en los dispositivos móviles independientemente de la plataforma, se ha llegado a utilizar con mucha frecuencia en programas nocivos móviles, ya que J2ME funciona en casi todos los móviles y smartphones. Desde 2006, el software malicioso J2ME ha llegado a ocupar el segundo lugar entre todos los objetos detectados por Kaspersky Lab con un 35%, cediendo el liderazgo a Symbian que tiene el 49%. Los autores de la investigación, Alexander Gostev, director del centro global de investigaciones de la compañía, y Denis Maslennikov, director del grupo de investigaciones de amenazas móviles, describen las nuevas, pero ya populares, tecnologías y métodos de los creadores de virus para móviles, como la copia de sí mismos en las tarjetas de memoria, la descarga de módulos complementarios desde Internet, las acciones espía, el deterioro de los datos del usuario, el polimorfismo y la cancelación de los instrumentos de protección incorporados en el SO.

El comportamiento más frecuente de los programas maliciosos para móviles observados durante los últimos dos años, es el envío de SMS a través del móvil del usuario, sin que éste lo sepa. Este tipo de programa, denominado Troyano-SMS, obliga al usuario a pulsar el botón y, de este modo, mandar un mensaje a un número determinado. Los creadores de estos programas usan métodos de engaño muy sofisticados. El canal más habitual de difusión de estos programas son los

portales WAP, donde se ofrece la descarga de software de distintos tipos y de contenidos multimedia. La mayoría de los Troyanos-SMS se disfraza de aplicaciones que ofrecen servicios gratuitos de intercambio de SMS o de acceso a Internet, así como de contenido de carácter erótico o pornográfico.

En el estudio también se describen las vulnerabilidades de los celulares más frecuentes, así como algunos de los incidentes virales más destacados en la actualidad, que demuestran que el software malicioso para móviles, al igual que los virus para las PC, tiene como objetivo, cada vez con mayor frecuencia, blancos concretos y locales, más que la propagación de epidemias.

Investigadores de diferentes instituciones europeas, coordinados por el Laboratorio de Sistemas Distribuidos de la Facultad de Informática de la UPM, elaboran una plataforma de desarrollo de servicios que, una vez finalizada, será capaz de procesar millones de datos por segundo, frente a las decenas de miles de datos por segundo que se pueden procesar con las tecnologías actuales.

Con esta tecnología se podrá combatir en tiempo real el fraude realizado con tarjetas de crédito, la duplicación de tarjetas SIM de telefonía móvil e incluso la realización fraudulenta de llamadas telefónicas no cobradas, entre otras muchas aplicaciones.

En el caso de la telefonía móvil, la duplicación de tarjetas SIM o el uso fraudulento de las líneas actualmente se detecta a posteriori, con el consecuente perjuicio económico. La nueva tecnología conseguirá asimismo en este caso que el fraude contra una operadora y/o un usuario de telefonía móvil no llegue a consumarse, evitando así el posible fraude de millones de euros y las molestias asociadas a los usuarios (presentación de reclamaciones, cancelación de tarjeta prepago, etc.).

La plataforma de desarrollo de servicios en tiempo real está siendo desarrollada en el seno del proyecto europeo Stream (Scalable Autonomic Streaming Middleware) financiado por el séptimo Programa Marco (7PM) de la Unión Europea. Dotado de una financiación europea de más de 3,5 millones de euros, se encuentra en la actualidad en el ecuador de su trayectoria, estando prevista su finalización en 2010.

El objetivo final del proyecto es conseguir una plataforma para el procesamiento de flujos masivos de datos en tiempo real. La principal novedad tecnológica es que Stream podrá emplear grandes clusters de nodos para procesar volúmenes masivos de datos, del orden de millones de datos por segundo. Las tecnologías actuales, basadas en el uso de nodos individualizados, todavía tienen una capacidad de procesamiento dos órdenes de magnitud inferior a las que aportará Stream.

El laboratorio de la Facultad de Informática de la UPM, que dirige el investigador Ricardo Jiménez-Peris, además de coordinar el proyecto europeo, es el encargado de desarrollar el procesador escalable de flujos de datos, que es el núcleo duro de Stream. Para ello paraleliza los operadores de consulta, pudiendo desplegar cada operador en un cluster de 100 nodos, lo que multiplica por 100 el volumen de datos que pueden procesarse.

El proyecto se enmarca en las iniciativas de cloud computing o computación en nube, que permite ofrecer servicios de computación a través de Internet, de tal forma que los usuarios pueden acceder a los servicios disponibles "en la nube de Internet" sin ser expertos en la gestión de los recursos que usan. Stream está diseñado para desplegarse en un entorno cloud computing, con características tales como la elasticidad, esto es, aumentar o disminuir el número de nodos automáticamente, según las necesidades de computación de cada momento, evitando el sobre-provisionamiento. La elasticidad permite asimismo reducir el coste del procesamiento al mínimo necesario, lo que resulta esencial en cloud computing, donde se paga por uso. Además de la Universidad Politécnica de Madrid, participan en la investigación, entre otros, Telefónica I+D, que contribuye con un sistema antifraude para la telefonía móvil, y la empresa griega Exodus, subsidiaria del banco de Pireos, que aplicará los resultados del proyecto a sus sistemas antifraude para pago con tarjetas de crédito.

Por otra parte, en la carrera de Informática se encuentran varias tesis referidas a la aplicación de redes neuronales artificiales en distintos campos como ser:

- "Clasificación de especies vegetales usando redes neuronales". (Admed Candí Torres). Diseña un método en base a un modelo de entidad artificial para la clasificación de especies vegetales que proporcione un mejor proceso de la información.
- "Reconocimiento de los rasgos faciales de una persona aplicando redes neuronales". (Maria Lourdes Velarde Flores). Propone y desarrolla una red neuronal para el reconocimiento de una persona a partir de los rasgos faciales contenidos en una fotografía, de la forma automática para obtener información necesaria acerca de está.
- "Seguridad en Internet utilizando redes neuronales". (José Alfredo Orozco Celis). Desarrolla un método de red neuronal para la detección de intrusos por maltrato en Internet.

1.3. PLANTEAMIENTO DEL PROBLEMA

Para el estudio del presente trabajo se identificaron diversos problemas, citados en la siguiente tabla. (Ver tabla 1.).

Tabla 1 Relación Causa - Efecto

CAUSA	EFECTO
La tecnología en telecomunicaciones avanza	Las formas de realizar fraude están constante - mente cambiando
Existen personas que obtienen un Servicio y no lo pagan.	Organizaciones pierden mucho dinero
Existe clonación en teléfonos celulares	Dos aparatos diferentes utilizando la misma cuenta
Falta de encriptación de datos en los canales de comunicación	Existe poca seguridad en la red móvil de comunicaciones
Ciberdelincuentes que cometen fraude	La pérdida anual en la industria de telecomuni - caciones

Fuente: Elaboración Propia

1.3.1. FORMULACION DEL PROBLEMA

¿Las Red Neuronal será capaz de detectar el fraude mediante cambios de consumo de la telefonía celular mediante la construcción de estructura de datos que representen el consumo histórico de cada uno de los usuarios?

1.4. HIPOTESIS

Considerando el principal problema del actual trabajo de investigación se plantea la siguiente hipótesis:

"La Red SOM, será capaz de clasificar las llamadas de telefonía celular para construir perfiles de usuario que representen su consumo, mediante el cual se detectara los cambios de comportamiento."

1.5. OBJETIVOS

1.5.1. OBJETIVO GENERAL

"Construir e implementar un Modelo de Red Neuronal con aprendizaje no supervisado para la detección de fraude en la telefonía móvil, para generar perfiles de usuario y detectar cambios de comportamiento".

1.5.2. OBJETIVOS ESPECIFICOS

- Construir un modelo de Red neuronal basada en la arquitectura de las redes som.
- Construir perfiles de usuario
- Implementar el modelo de Red neuronal
- Detectar cambios de comportamiento.

1.6. JUSTIFICACION

1.6.1. JUSTIFICACION CIENTIFICA

El estudio conjunto de las redes neuronales artificiales y la necesidad creciente de consolidar una privacidad y seguridad, es de por si bastante complejo ya que se requiere un dominio total de ambas materias como es de redes e inteligencia artificial. Así como también el crecimiento de las formas de fraude y la complejidad de estos fraudes hacen que la inteligencia artificial más específicamente las redes neuronales artificiales sean una disyuntiva de estudio e investigación, lo cual sugiere nuevos caminos de indagación en este campo, dado el amplio abanico de posibilidades que ofrece este tipo de estudio.

1.6.2. JUSTIFICACION SOCIAL

Los beneficios sociales que se pueden obtener son muy amplios para la sociedad el cual es el directo beneficiario, al implantar este prototipo.

Muchas personas están expuestas a fraudes informáticos, al realizar llamadas telefónicas mediante su celular. Es por esta razón que el presente trabajo muestra una solución para poder detectar dicho fraude.

1.6.3. JUSTIFICACION ECONOMICA

Económicamente permitirá a los usuarios tener mayor seguridad en sus actividades para así poder trabajar, también resultara una inversión conveniente el desarrollo de un software basados en redes neuronales para la detección de fraude en teléfono móviles.

1.7. METODOLOGIA

Se realizaran experimentos que se dividirán en dos partes: la primera se enfocara en el entrenamiento de la red y la generación de los patrones para construir posteriormente los perfiles de usuario; la segunda prueba se enfocara en el análisis de las llamadas de los usuarios con alto consumo y el correspondiente análisis.

1.8. ALCANCES Y APORTES

1.8.1. ALCANCES

Se construirá un modelo de Red Neuronal para la detección de fraude en telefonía celular, y se realizara una distribución de frecuencia para analizar los cambios de comportamiento en el consumo de los usuarios.

El presente trabajo se encuentra orientado a la construcción de un modelo neuronal y su implementación en un software neuronal.

Se probará el adecuado funcionamiento y el rendimiento del modelo con datos.

1.8.2. APORTES

Con este trabajo se quiere demostrar, que es posible aplicar la capacidad de las Redes Neuronales artificiales para la detección de fraude en telefonía celular.

Se construirá y se implementara un modelo de Red Neuronal en un software neuronal.

Será una herramienta útil para poder detectar fraude en teléfonos celulares.

La detección de fraudes a teléfonos celulares, podrá ayudar a empresas de telefonías celulares a evitar dichos fraudes mediante los cambios de consumo que realicen.

CAPITULO II
MARCO TEORICO

CAPITULO II

MARCO TEORICO

2.1. DEFINICION DE FRAUDE EN TELEFONIA CELULAR

Fraude se puede describir de una manera simple como "cualquier actividad por la cual un servicio es obtenido sin la intención de pagarlo". [Gosset & Hyland, 1999]. Muchas veces las organizaciones calculan cuánto dinero pierden debido al fraude definiéndolo como "el dinero que se pierde en clientes/cuentas por los cuales no se recibe ningún pago" [Gosset & Hyland, 1999]. Sin embargo, para los fines de detección, tal definición no es apropiada debido que el fraude solo sería detectado una vez que ha ocurrido. De hecho, especificar qué es el fraude puede ser muy difícil, debido a que la diferencia entre un comportamiento fraudulento y uno que no lo es puede ser muy pequeña; por lo tanto lo más prudente es clasificar al fraude en diferentes tipos y describir cada uno de ellos.

2.1.1. CLASIFICACION DE TIPOS DE FRAUDE

A continuación se presentan diferentes tipos de fraude que deben ser tomados en cuenta cuando se estudia este problema.

2.1.2. FRAUDE CONTRACTUAL

Todos los fraudes en esta categoría generan a priori ganancia para la empresa a través del uso normal de los teléfonos celulares, pero finalmente el usuario no tiene intenciones de pagar por el servicio que se le brindó. Un ejemplo de este tipo de fraude es el denominado *por suscripción*. El mismo puede tomar varios matices, pero puede ser dividido principalmente en dos casos: 1) aquel donde el usuario contrata el servicio sin la intención de pagarlo nunca; 2) luego de varias facturaciones el usuario toma la decisión de no pagar por el uso del mismo. Este último caso usualmente resulta en un *cambio dramático de su comportamiento en el uso del servicio* y será el caso modelo que utilizaremos para nuestro trabajo. De todas maneras, el primer caso no puede ser detectado a través de información de uso, ya que la misma no existe cuando comienza a utilizar el servicio y es

necesaria información adicional tal como su condición crediticia para analizar el riesgo que implique darle el servicio a un determinado usuario.

2.1.3. FRAUDE POR VIOLACION DE SEGURIDAD

Todos los fraudes en esta categoría le permiten, a quién logra ingresar en sistemas inseguros, brindar de manera ilegal servicios a terceros. Es decir, utilizar recursos de la compañía de manera desleal. Ejemplos de tales fraudes son el fraude hacia una PABX (Private Automatic Branco Exchange – Central telefónica que provee acceso a diferentes servicios de comunicaciones como conexión a otras redes de telecomunicaciones [ITS, 2000]) y el ataque a la red.

En el fraude hacia una PABX el "atacante" llama repetidamente a la misma, tratando de tener acceso a una línea externa; una vez que se tiene acceso, pueden realizar llamadas salientes de alto valor (nacionales o internacionales de larga duración) simplemente pagando un precio de acceso a la PABX. Usualmente, tales ataques están asociados con el uso de teléfonos clonados, de manera que ni siquiera pagan los precios de acceso.

En los ataques a la red, se intenta ingresar a las redes de computadoras a través de módems que se configuran en las mismas para poder realizar tareas remotas de administración y soporte. Una vez que se accede por uno de ellos, el atacante intenta ingresar a la red y configurar ciertos equipos para su propio beneficio. Estos fraudes se caracterizan por llamadas cortas y continuas al mismo número en el caso de fraude a una PABX o llamadas cortas a números secuenciales en el caso de fraude de red, por lo cual es este el comportamiento que debe ser detectado.

2.1.4. FRAUDE TECNICO

Todos los fraudes en esta categoría involucran ataques contra las debilidades de la tecnología de los sistemas de telefonía celular (móvil). Tales fraudes típicamente necesitan habilidad y algún conocimiento técnico inicial, aunque una

vez que se encontró una debilidad esta información se distribuye rápidamente de manera que gente sin los conocimientos necesarios pueda usarla.

Ejemplos de este tipo de fraude son la clonación de teléfonos y el fraude interno técnico. En una clonación, los parámetros de autenticación de un móvil son copiados a otro equipo, de modo que la red "crea" que es el teléfono original quien esta intentando validarse.

En un fraude interno técnico, empleados de la compañía pueden alterar cierta información en los equipos de comunicaciones para permitir a ciertos usuarios reducir el costo de acceso a los diferentes servicios. El comportamiento de uso de estos clientes depende de cuánto tiempo desean permanecer sin ser detectados. En la situación en la que el atacante cree que puede "esconderse" por un largo tiempo, deberá no salirse del comportamiento normal de uso para no ser detectado. Si cambiara su estilo de uso (promedio de llamadas locales, nacionales, internacionales) la solución que propondremos en este trabajo lo encontraría rápidamente. En general, este tipo de fraude es de corta duración ya que se intenta hacer uso del servicio lo máximo posible hasta ser detectado y cortado el servicio.

2.1.5. FRAUDE DE PROCEDIMIENTO

Todos los fraudes que se describen en esta sección implican la intención de evitar los procedimientos implementados para detener el fraude. A menudo estos ataques se enfocan en las debilidades de los procedimientos de negocio usados para dar acceso a los sistemas.

Un ejemplo típico de este fraude es el de Roaming (utilizar el teléfono en otra red, ejemplo otro país, para luego cobrarse el uso en el país de origen). En este caso, el procedimiento de facturación generalmente se realiza unos días después que las llamadas fueron realizadas, cuando el suscriptor puede ya no existir. Sin embargo este tipo de acciones son previstas por casi todos los sistemas de facturación en telefonía celular.

Otro ejemplo es el de intentar registrarse en la compañía con datos falsos para lo cual los procesos administrativos deben ser controlados y revisados constantemente para evitar el ingreso de "falsos" clientes a la red.

2.2. DETECCION DE FRAUDE EN TELEFONIA CELULAR

Cuando se inicia una llamada de celular, las celdas o switches registran que la misma se está realizando y producen información referida a este evento. Estos registros de datos son comúnmente llamados CDR's (Call Detail Records). Los CDR's contienen importante información sobre la llamada para que luego ésta pueda ser cobrada a quien corresponda [ASPeCT, 1997].

Estos registros también pueden ser usados para detectar actividad fraudulenta considerando indicadores de fraude bien estudiados. Es decir, procesando una cantidad de CDR's recientes y comparando una función de los diferentes campos tales como IMSI (International Mobile Subscriber Identity, que identifica unívocamente un usuario en una red de telefonía celular), fecha de la llamada, hora de la llamada, duración, tipo de llamada con un cierto criterio determinado [Moreau & Preneel, 1997]. Si esta función devuelve un valor que se considera fuera de los límites normales, se activa una alarma, que debe ser tomada en cuenta por los analistas de fraude para constatar si realmente hubo o no actividad de mala fe. Para poder procesar estos CDR's es necesario realizar previamente un proceso conocido en telecomunicaciones como *mediación*, en el cual se lee la información con el formato de registro en el que vienen los CDR's (el mismo puede ser de longitud variable dependiendo del tipo de llamada y del proveedor del switch) y se codifica en un nuevo formato de registro entendible por el sistema de fraude en este caso.

Los sistemas existentes de detección de fraude intentan consultar secuencia de CDR's comparando alguna función de los campos con criterios fijos conocidos como *Triggers*. Un *trigger*, si es activado, envía una alarma que lleva a la investigación por parte de los analistas de fraude. Estos sistemas realizan lo que se conoce como *Análisis absoluto de CDR's* y son buenos para detectar los extremos de la actividad fraudulenta. En cambio, para realizar un *análisis*

diferencial, se monitorean patrones de comportamiento del teléfono celular comparando sus más recientes actividades con la historia de uso del mismo. Un cambio en el patrón de comportamiento es una característica sospechosa de ser un escenario fraudulento [ASPeCT, 1997].

La figura 2.1, que se presenta a continuación, muestra las diferencias entre el análisis absoluto y el análisis diferencial.

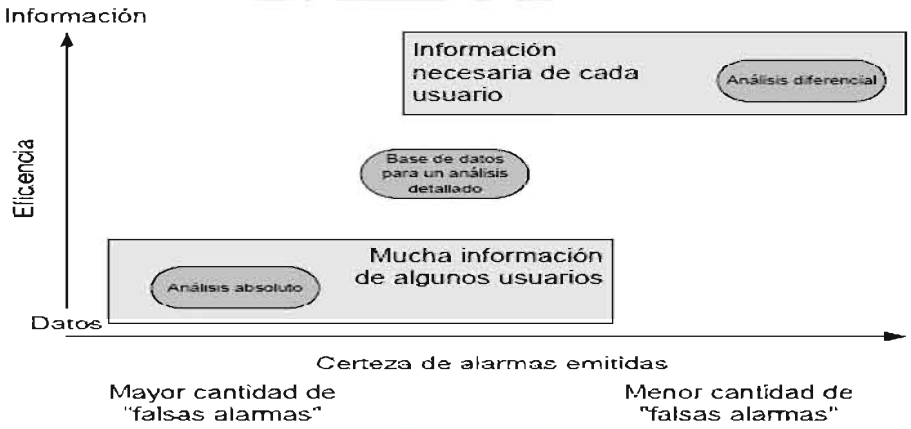


Figura 2.1: Análisis Absoluto vs. Análisis Diferencial
Fuente: [ASPeCT, 1997].

Se puede observar en la figura 2.1 que utilizando el análisis absoluto solo se pueden analizar algunos usuarios cuyo consumo supere cierto nivel, para lo cual es necesario tener mucha información del mismo. Además la certeza en las alarmas que el sistema emitirá no es completamente confiable ya que pueden existir muchos casos en los cuales se clasifique a un usuario como fraudulento cuando realmente no lo es (falsas alarmas). En cambio en el análisis diferencial la certeza de obtener mejores resultados aumenta permitiendo además poder analizar a cada uno de los usuarios.

A su vez dentro del análisis diferencial hay diferentes enfoques en la detección de fraude [Gosset & Hyland, 1999]:

• *El enfoque por enseñanza*: el cual se tipifica por el uso de redes neuronales supervisadas o herramientas de detección de fraude basadas en reglas. A estas herramientas se les presentan casos de fraude existentes y luego tratan de encontrar indicios de fraude basado en lo que han aprendido o "se les enseñó". El enfoque por enseñanza es útil para detectar fraude por suscripción y violación de seguridad. Adicionalmente, una vez que se descubre fraude técnico, también puede, posteriormente, detectarse el mismo utilizando este enfoque.

• *El enfoque por aprendizaje*: en el cual generalmente se tipifica por el uso de redes neuronales no supervisadas donde la herramienta de detección de fraude aprende por sí sola cuál es el comportamiento esperado del usuario. Es muy útil para detectar cambios de comportamiento y por lo tanto más eficiente en la detección de fraude por suscripción y violación de seguridad.

2.2.1. ENFOQUE POR ENSEÑANZA

En este enfoque, es necesario tener ejemplos reales de fraude. Estos ejemplos son usados para "enseñar" a la herramienta qué es lo que debe buscar. En el caso de un sistema basado en reglas, los ejemplos son analizados por sus componentes de fraude que luego se traducen en reglas que utilizan umbrales o medidas relativas. En el caso de las redes neuronales supervisadas se usan los ejemplos de fraude y los ejemplos de usuarios no fraudulentos para enseñarle a la herramienta cuáles comportamientos son buenos y cuáles no lo son. Ambos tipos de herramientas deberían identificar comportamientos de alguna manera similar a los ejemplos de fraude usados o a los ejemplos de buen comportamiento; si identifican algún comportamiento como "parecido" al de un ejemplo de fraude, deben emitir una alarma.

2.2.2. ENFOQUE POR APRENDIZAJE

En este enfoque, la herramienta aprenderá el comportamiento típico de un usuario y emitirá una alarma cuando este comportamiento haya cambiado sensiblemente.

La habilidad de la herramienta para monitorear el comportamiento de los usuarios la hace muy útil para detectar fraudes de los que no se sabe nada como así todos los casos de fraude por suscripción, que resultan en cambios de comportamiento. Si se sabe poco acerca del fraude existente en el sistema, esta es una buena forma de trabajar y obtener buenos ejemplos de comportamiento fraudulento; sin embargo, hay algunos puntos importantes a tener en cuenta cuando se utiliza este enfoque entre los cuales se puede destacar que no es posible enseñarle a esta herramienta qué buscar y si los parámetros de evolución no se configuran correctamente, puede llegar a fallar y no detectar cambios de comportamiento que lancen las alarmas correspondientes.

Con las redes neuronales no supervisadas se pueden crear perfiles de usuario basados en su comportamiento reciente y compararlo con su consumo histórico que evoluciona a través del tiempo con las llamadas realizadas.

Nuestro trabajo se centrará en la construcción de una herramienta que utilice este enfoque ya que es muy difícil encontrar a priori un escenario en el cual se conozcan muchos casos de fraude para utilizar el enfoque por enseñanza.

2.3. REDES NEURONALES

Las *Redes Neuronales Artificiales* son redes de elementos simples interconectadas masivamente en paralelo y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico [Kohonen, 1988]. La compleja operación de las redes neuronales es el resultado de abundantes lazos de realimentación junto con no linealidades de los elementos de proceso y cambios adaptativos de sus parámetros, que pueden definir incluso fenómenos dinámicos muy complicados [Hilera González & Martínez Hernando, 2000].

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante [Hilera González & Martínez

Hernando, 2000]. Las principales ventajas ofrecidas por las mismas son:

- *Aprendizaje adaptativo*: Capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.
- *Autoorganización*: Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- *Tolerancia a fallos*. La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
- *Operación en tiempo real*: Los computadores neuronales pueden ser realizados en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad [Maren, 1990].

Basados en esta definición las redes neuronales son capaces de agrupar las llamadas y clasificarlas de una manera acorde y construir, basados en esta clasificación, perfiles de usuario que representen su consumo y así luego detectar los cambios de comportamiento.

2.4. INTRODUCCION A LAS REDES NEURONALES ARTIFICIALES

En los últimos años de exploración en inteligencia artificial, los investigadores se han intrigado por las redes neuronales. Como su nombre lo implica, una red neuronal artificial consiste en una red de neuronas artificiales interconectadas. El concepto se basa vagamente en cómo pensamos que funciona el cerebro de un animal. Un cerebro consiste en un sistema de células interconectadas, las cuales son, aparentemente, responsables de los pensamientos, la memoria y la conciencia. Las neuronas se conectan a muchas otras neuronas formando uniones llamadas sinapsis; las señales electroquímicas se propagan de una neurona a otra a través de estas sinapsis. Las neuronas demuestran plasticidad: una habilidad de cambiar su respuesta a los estímulos en el tiempo, o aprender; en una red neuronal artificial, se imitan estas habilidades por software [Hilera González & Martínez Hernando, 2000].

2.4.1. ELEMENTOS DE UNA RED NEURONAL ARTIFICIAL

Cualquier modelo de red neuronal consta de dispositivos elementales de proceso: las *neuronas*. A partir de ellas, se puede generar representaciones específicas de tal forma que un estado conjunto de ellas pueda significar una letra, un número o cualquier otro objeto. La neurona artificial pretende mimetizar las características más importantes de las neuronas biológicas. Cada neurona i-ésima está caracterizada en cualquier instante por un valor numérico denominado *valor o estado de activación* $a_i(t)$; asociado a cada unidad existe una *función de salida*, f_i, que transforma el estado actual de activación en una *señal de salida* y_i. Dicha señal es enviada a través de los canales de comunicación unidireccionales a otras unidades de la red; en estos canales la señal se modifica de acuerdo con la sinapsis (el *peso*, w_{ji}) asociada a cada uno de ellos según una determinada regla. Las señales moduladas que han llegado a la unidad j-ésima se combinan entre ellas generando así la *entrada total*, $\mathrm{Net_j}$:

$$Net_j = \sum_i y_i w_{ji}$$

Una *función de activación*, F, determina el nuevo estado de activación $\mathrm{a_j(t+1)}$ de la neurona, teniendo en cuenta la entrada total calculada y el anterior estado de activación $\mathrm{a_j(t)}$. Si se tienen N unidades (neuronas), se puede ordenarlas arbitrariamente y designar la j-ésima unidad como Uj. Su trabajo es simple y único, y consiste en recibir las entradas de las células vecinas y calcular un valor de salida, el cual es enviado a todas las células restantes.

En cualquier sistema de redes neuronales que se esté modelando, es útil caracterizar tres tipos de unidades:

- *Entradas*: estas unidades reciben desde el entorno.
- *Salidas*: estas unidades envían la señal fuera del sistema (salidas de la red).
- *Ocultas*: son aquellas cuyas entradas y salidas se encuentran dentro del sistema; es decir que no tienen contacto con el exterior.

Se conoce como *capa* o *nivel* a un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino.

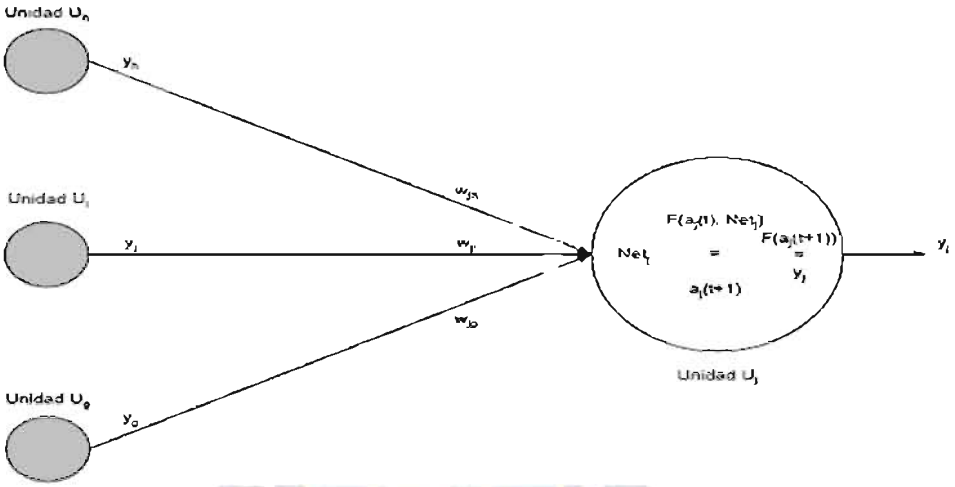


Figura 2.2: Estados de una neurona

Fuente: [Hilera González & Martínez Hernando, 2000]

La figura 2.2 muestra cómo se compone el estado de una neurona basado en los valores que le "llegan" de cada una de las neuronas de la capa anterior.

En la mayoría de los casos, F *es la función identidad*, por lo que el estado de activación de una neurona en t+1 coincidirá con el Net de la misma en t. Según esto, la salida de una neurona quedará según la expresión:

$$y_i(t+1) = f(Net_i) = f\left(\sum_{j=1}^{N} w_{ij} y_j(t)\right)$$

Además, normalmente la función de activación no está centrada en el origen del eje que representa el valor de la entrada neta, sino que existe cierto desplazamiento debido a las características internas de la propia neurona y que no es igual en todas ellas. Este valor se denota como θ_i y representa el umbral de activación de la neurona i.

$$y_i(t+1) = f(Net_i - \theta_i) = f\left(\sum_{j=1}^{N} w_{ij} y_j(t) - \theta_i\right)$$

2.4.2. TOPOLOGIA DE LAS REDES NERONALES ARTIFICIALES

La topología o arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de

neuronas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: *el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas*.

En las redes monocapa (1 capa) se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red. También pueden existir *conexiones autorrecurrrentes* (salida de una neurona conectada a su propia entrada).

Las redes multicapa son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Normalmente, todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a las entradas de la red, y envían señales de salida a una capa posterior, más cercana a la salida de la red; a estas conexiones se les denomina *conexiones hacia adelante* o *feedforward*. Sin embargo, en un gran número de estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de las capas anteriores, a estas conexiones se las denomina *conexiones hacia atrás* o *feedback*.

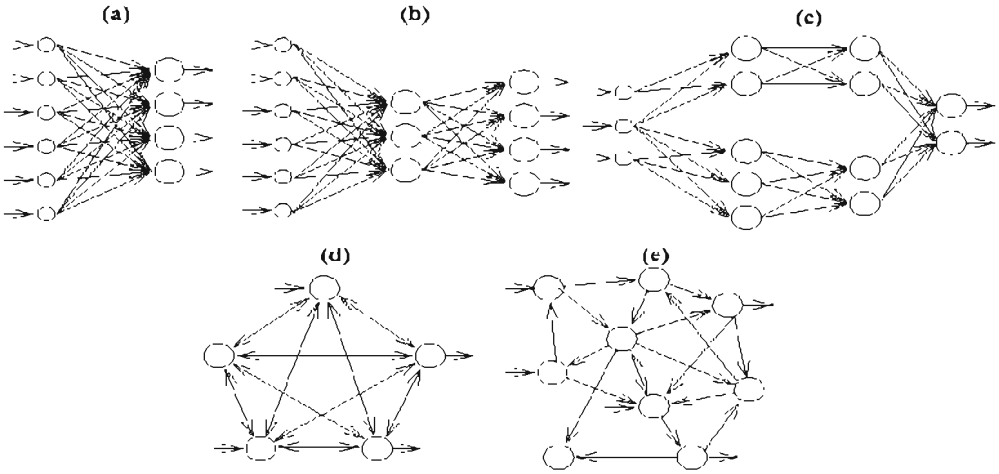


Figura 2.3: Topologías de redes neuronales

Fuente: [Hilera González & Martínez Hernando, 2000]

En la figura 2.3 podemos visualizar 5 topologías de redes diferentes: (**a**) Un Perceptrón de una capa (SLP) conectado completamente. (**b**) Un Perceptrón multicapa (MLP) conectado completamente. (**c**) Un MLP modular. (**d**) Una red recurrente conectada completamente. (**e**) Una red recurrente conectada parcialmente.

2.4.3. MECANISMO DE APRENDIZAJE

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables o el margen de error es menor o igual al que se ha definido como aceptable.

Un aspecto importante respecto al aprendizaje en las redes neuronales es el conocer cómo se modifican los valores de los pesos; es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red *aprenda* una nueva información. Estos criterios determinan lo que se conoce como la *regla de aprendizaje* de la red. De forma general, se suelen considerar dos tipos de reglas: las que responden a lo que habitualmente se conoce como aprendizaje supervisado, y las correspondientes a un aprendizaje no supervisado. La diferencia fundamental entre ambos tipos de aprendizaje está en la existencia o no de un agente externo (*supervisor*) que controle el proceso de aprendizaje de la red.

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede *aprender* durante su funcionamiento habitual o si el aprendizaje supone la *desconexión* de la red; es decir su inhabilitación hasta

que el proceso termine. En el primer caso, se trata de un aprendizaje ON LINE, mientras que el segundo es lo que se conoce como aprendizaje OFF LINE.

En las redes con aprendizaje ON LINE no se distingue entre fase de entrenamiento y de operación, de tal forma que los pesos varían dinámicamente siempre que se presente una nueva información al sistema.

Cuando el aprendizaje es OFF LINE, se distingue entre una *fase de aprendizaje o entrenamiento* y una *fase de operación o funcionamiento*, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de test o prueba que serán utilizados en la correspondiente fase. En estas redes, los pesos de las conexiones permanecen fijos después que termina la etapa de entrenamiento de la red.

2.4.3.1. REDES CON APRENDIZAJE SUPERVISADO

La técnica mayormente utilizada para realizar un aprendizaje supervisado consiste en ajustar los pesos de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, una función de error cometido en la salida.

Existen varias formas de calcular el error y luego adaptar los pesos con la corrección correspondiente. Una de las más implementadas utiliza una función que permite cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanto más información se tenga del error cometido, más rápido se puede aprender [Widrow & Hoff, 1960]. El error medio se expresa por la ecuación:

$$Error_{global} = \frac{1}{2P}\sum_{k=1}^{P}\sum_{j=1}^{N}\left(y_j^{(k)} - d_j^{(k)}\right)^2$$

Dónde:

N = Número de neuronas de salida.

P = Número de informaciones que debe aprender la red.

d_j = Valor de salida deseado para la neurona j.

y_j = Valor de salida obtenido para la neurona j.

k = patrón k-ésimo presentado a la red.

Por lo tanto, de lo que se trata es de encontrar unos pesos para las conexiones de la red que minimicen esta función de error. Para ello, el ajuste de los pesos de las conexiones de la red se puede hacer de forma proporcional a la variación relativa del error que se obtiene al variar el peso correspondiente:

$$\Delta w_{ji} = k \frac{\partial Error_{global}}{\partial w_{ji}}$$

Dónde:

Δw_{ji} = Variación en el peso de la conexión entre las neuronas i y j.

Mediante este procedimiento, se llegan a obtener un conjunto de pesos con los que se consigue minimizar el error medio, con la presentación de cada nuevo patrón de entrenamiento a la red.

2.4.3.2. REDES CON APRENDIZAJE NO SUPERVISADO

Las redes con aprendizaje no supervisado no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no es correcta; por ello, suele decirse que estas redes son capaces de *auto organizarse*. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presentan en su entrada.

En algunos casos, la salida representa el grado de *familiaridad* o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado). En otro caso podría realizar una *clusterización* o establecimiento de *patrones* o *categorías*, indicando la red a la salida a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas. Una variación de esta categorización es el *prototipado*. En este caso, la red obtiene ejemplares o prototipos representantes de las clases a las que pertenecen las informaciones de entrada.

Finalmente, algunas redes con aprendizaje no supervisado lo que realizan es un *mapeo de características*, obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada, de tal forma que si se presentan a la red informaciones similares, siempre sean afectadas neuronas de salida próximas entre sí, en la misma zona del mapa.

2.4.4. CLASIFICACION DE REDES NEURONAES

Existen muchas formas de clasificar a los diferentes tipos de redes neuronales ya sea por su forma de aprendizaje, su topología, etc. Sin embargo, es interesante citar una clasificación que las divide en 3 categorías según su funcionamiento [Kohonen, 1995]:

- Redes de transferencia de señal.
- Redes de transición de estados.
- Redes con aprendizaje competitivo.

En las redes de transferencia de señal, la señal de entrada se transforma en una señal de salida.

La señal atraviesa la red y experimenta una transformación de algún tipo. Estas redes tienen usualmente un conjunto de funciones prefijadas, que se parametrizan. En las redes de transición de estados el comportamiento dinámico de la red es esencial. Dada una señal de entrada, la red converge a un estado estable, que, si se tiene éxito, corresponde a una solución del problema que se le presentó.

Finalmente, en las redes con aprendizaje competitivo, o *redes autorganizables*, todas las neuronas de la red reciben la misma señal de entrada; las celdas compiten con sus vecinas laterales y la que mayor actividad tiene "gana". El aprendizaje se basa en el concepto de la "neurona ganadora".

2.5. RED NEURONAL SUPERVISADA – EL PERCEPTRON

Este fue el primer modelo de red neuronal artificial desarrollado por Rosenblatt en 1958 [Rosenblatt, 1958]. Un Perceptrón, formado por varias neuronas lineales para recibir las entradas a la red y una de salida, es capaz de decidir cuándo una entrada presentada a la red y pertenece a una de las dos clases que es capaz de reconocer. Es una red que utiliza aprendizaje supervisado OFF LINE con conexiones hacia adelante.

La única salida del Perceptrón realiza la suma ponderada de las entradas, resta el umbral y pasa el resultado a una función de transferencia de tipo escalón. La regla de decisión se basa en responder +1 si el patrón presentado pertenece a la clase A, o –1 si el mismo pertenece a la clase B. La salida dependerá de la entrada neta (suma de las entradas xi ponderadas) y del valor umbral θ.

Sin embargo, este modelo de red neuronal no tiene demasiadas aplicaciones ya que solo puede clasificar las entradas en solo dos grupos diferentes; es por ello que se utiliza el Perceptrón multicapa que contiene varias capas de neuronas de entre la entrada y la salida de la misma. Esta red permite establecer regiones de decisión mucho más complejas.

El Perceptrón básico de dos capas sólo puede establecer dos regiones separadas por una frontera lineal en el espacio de patrones de entrada; un Perceptrón multicapa puede formar cualquier región convexa en este espacio. La regla de aprendizaje utiliza una técnica de corrección de error como la explicada anteriormente y consiste en: 1) inicicializar los pesos de la red con valores aleatorios, 2) presentación de un patrón de entrada y propagación de los valores hasta calcular la salida, 3) adaptar los pesos basados en el error cometido teniendo en cuenta la salida esperada. Este procedimiento se realiza hasta que el error obtenido es menor o igual al error aceptado.

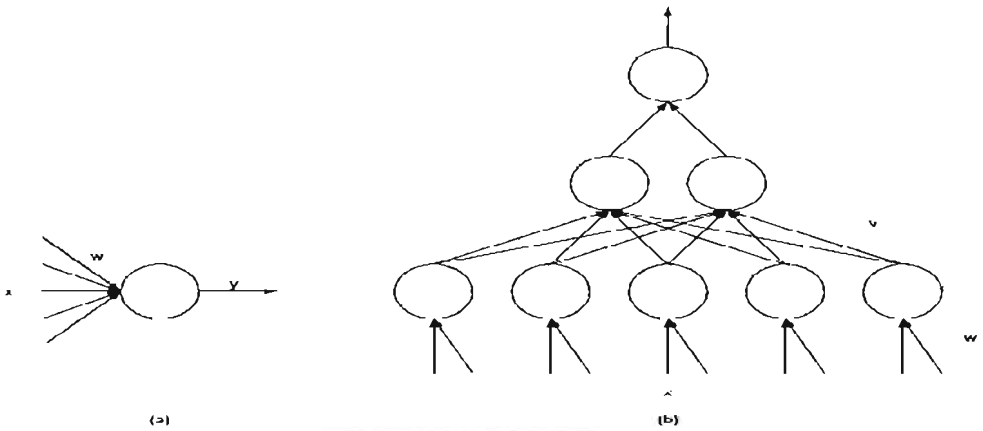


Figura 2.4: Un Perceptrón Multicapa

Fuente: [Rosenblatt, 1958]

La figura 2.4 muestra en **(a)** un Perceptrón simple y en **(b)** un Perceptrón multicapa. Ambos utilizan la misma técnica de aprendizaje, el segundo es capaz de clasificar la información de una mejor manera que el primero.

2.6. SELF ORGANIZING MAPS (SOM)

Existen evidencias que demuestran que en el cerebro hay neuronas que se organizan en muchas zonas, de forma que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de *mapas bidimensionales* [Beveridge, 1996]. Por ejemplo, en el sistema visual se han detectado mapas del espacio visual en zonas del córtex (capa externa del cerebro); también en el sistema auditivo se detecta una organización según la frecuencia a la que cada neurona alcanza mayor respuesta [Hilera González & Martínez Hernando, 2000].

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje. Esto sugiere, por tanto, que el cerebro podría poseer capacidad inherente de formar *mapas topológicos* de las informaciones recibidas del exterior.

A partir de estas ideas, Teuvo Kohonen presentó en 1982 [Kohonen, 1982] un sistema con un comportamiento semejante; se trata de un modelo de red neuronal con capacidad para formar *mapas de características* de manera similar a como

ocurre en el cerebro. El objetivo de Kohonen era demostrar que un estímulo externo (información de entrada) por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de mapas. Estudiaremos, entonces, este modelo llamado *Self Organizing Maps (SOM)* que se basa en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red. Este modelo es uno de los más populares que se utilizan en redes neuronales artificiales y pertenece a la categoría de redes con aprendizaje competitivo.

2.6.1. ALGORITMO DEL SOM

El algoritmo de aprendizaje del SOM está basado en el aprendizaje no supervisado y competitivo, lo cual quiere decir que no se necesita intervención humana durante el mismo y que se necesita saber muy poco sobre las características de la información de entrada. Podríamos, por ejemplo, usar un SOM para clasificar datos sin saber a qué clase pertenecen los mismos [Hollmen, 1996]. El mismo provee un mapa topológico de datos que se representan en varias dimensiones utilizando unidades de mapa (las neuronas) simplificando el problema [Kohonen, 1995]. Las neuronas usualmente forman un mapa bidimensional por lo que el mapeo ocurre de un problema con muchas dimensiones en el espacio a un plano [Hollmen, 1995]. La propiedad de preservar la topología significa que el mapeo preserva las distancias relativas entre puntos [Kohonen, 1982]. Los puntos que están cerca unos de los otros en el espacio original de entrada son mapeados a neuronas cercanas en el SOM; por lo tanto, el SOM sirve como herramienta de análisis de clases de datos de muchas dimensiones [Vesanto & Alhoniemi, 2000]; además tiene la capacidad de *generalizar* [Essenreiter, Karrenbach & Treitel, 1999], lo que implica que la red puede reconocer o caracterizar entradas que nunca antes ha encontrado; una nueva entrada es asimilada por la neurona a la cual queda mapeada.

El SOM es un vector bidimensional de neuronas:

$$M = \{m_1, \ldots, m_{pxq}\}$$

Una neurona es un vector llamado patrón representado de la siguiente forma:

$$m_j = \{m_{j1}, \ldots, m_{jn}\}$$

La neurona tiene las mismas dimensiones que los vectores de entrada (datos de entrada), es decir que es n-dimensional. Las neuronas están conectadas a las neuronas adyacentes por una relación de vecinos. Esta dicta la topología, o la estructura, del mapa; usualmente, las neuronas están conectadas unas con otras en una topología hexagonal o rectangular. En la figura 2.4 podemos observar **(a)** una estructura rectangular y **(b)** una estructura hexagonal.

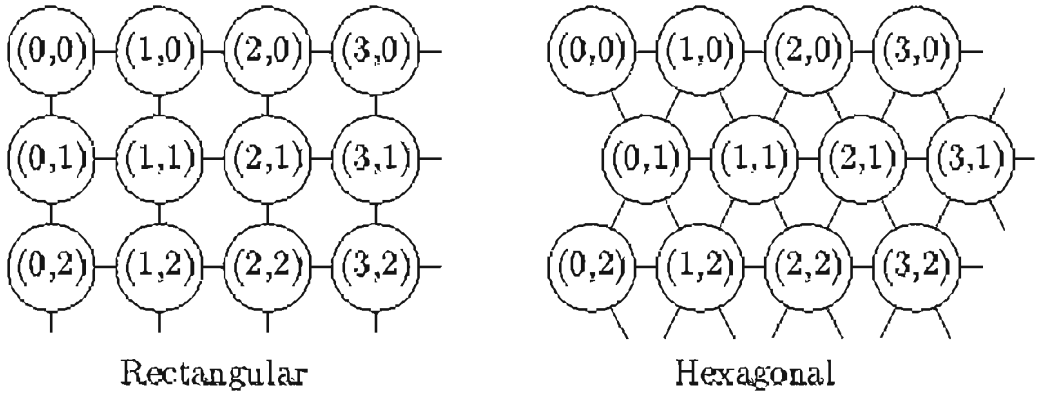


Figura 2.5: Estructuras de los mapas
Fuente: [Kohonen, 1995]

Se puede definir una distancia entre las neuronas de acuerdo a su relación de topología; las mismas pueden ser vecinas inmediatas (las neuronas adyacentes) que pertenecen al *vecindario* N_c de la neurona m_c. La función de vecindario es una función decreciente en el tiempo: $N_c = N_c(t)$.

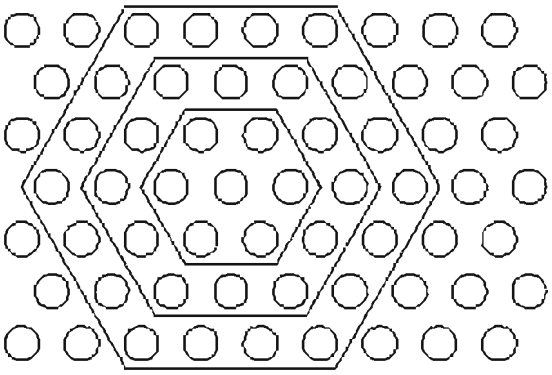


Figura 2.6: Vecindario de una neurona
Fuente: [Hollmen, 1995].

En la figura 2.6 podemos observar vecindarios de distintos tamaños. En el hexágono más pequeño se encuentran todas las neuronas vecinas que pertenecen al vecindario más pequeño de la neurona ubicada en el centro.

En el algoritmo básico del SOM, las relaciones topológicas y el número de neuronas *son fijos* desde el comienzo; este número de neuronas determina la escala o la granularidad del modelo resultante. La selección de la granularidad afecta la certeza y la capacidad de generalizar del modelo. Debe tenerse en cuenta que la granularidad y la generalización son objetivos contradictorios [Hollmen, 1995]. Mejorando el primero, se pierde en el segundo, y viceversa. Esto se debe a que si aumentamos el primero se obtendrán muchos más grupos para poder clasificar los datos de entrada, evitando que se pueda generalizar el espacio en clases más abarcativas. De manera inversa, si se generaliza demasiado se puede perder información que caracterice a un grupo específico que quede incluido en otro por la falta de granularidad.

2.6.1.1. PRE-PROCESAMIENTO DE LOS DATOS

Los datos que alimentan al SOM incluyen toda la información que toma la red. Si se le presenta información errónea, el resultado es erróneo o de mala calidad. Entonces, el SOM, tanto como los otros modelos de redes neuronales, deben eliminar la información "basura" para que no ingrese al sistema. Por lo cual se

debe trabajar con un subconjunto de los datos; estos deben ser relevantes para el modelo a analizar. También se deben eliminar los errores en los datos; si los mismos se obtienen a través de una consulta a una base de datos, el resultado puede incluir datos erróneos debido a la falta de integridad de la base; entonces estos deben ser filtrados usando conocimientos previos del dominio del problema y el sentido común.

Comúnmente los componentes de los datos de entrada se normalizan para tener una escala de 0 a 1. Esto asegura que por cada componente, la diferencia entre dos muestras contribuye un valor igual a la distancia medida calculada entre una muestra de entrada y un patrón. Es decir que los datos deben previamente codificarse (normalizarse). De lo contrario no será posible usar la distancia como una medida de similitud. Esta medida debe ser cuantificable por lo que la codificación debe ser armónica con la medida de similitud utilizada. La medida mayormente utilizada es la distancia Euclídea. Los datos simbólicos no pueden ser procesados por un SOM como tales, por lo que deben ser transformados a una codificación adecuada.

2.6.1.2. INICIALIZACION

Existen varios tipos de inicializaciones para los valores de las neuronas (patrones): entre ellos se pueden nombrar la inicialización al azar y la inicialización utilizando usando las primeras muestras. En la inicialización al azar se asignan valores aleatorios a los patrones; se utiliza cuando se sabe muy poco o nada sobre los datos de entrada en el momento de comenzar el entrenamiento. La inicialización utilizando las primeras muestras utiliza los primeros datos de entrada asignándolos a los patrones; tiene la ventaja que automáticamente se ubican en la parte correspondiente del espacio de entrada.

2.6.1.3. ENTRENAMIENTO

El entrenamiento es un proceso iterativo a través del tiempo. Requiere un esfuerzo computacional importante, y por lo tanto, consume mucho tiempo. Este consiste de muestras del conjunto de datos de entrada que van ingresando a la red para que la misma las "aprenda". El aprendizaje consiste en elegir una neurona

ganadora por medio de una medida de similitud y actualizar los valores de los patrones en el vecindario del ganador; este proceso se repite varias veces para poder ir refinando (acotando) el error y acercar las neuronas a una representación más adecuada de los datos de entrada.

En un paso del entrenamiento, un vector muestra se toma de los datos de entrada; este vector es presentado a todas las neuronas en la red y se calcula la medida de similitud entre la muestra ingresada y todos los patrones. La unidad más parecida o *Best Matching Unit (BMU)* se elige como el prototipo con la mayor similitud con la muestra de entrada; esta similitud usualmente se define con una medida de distancia vectorial. Por ejemplo, en el caso de la distancia Euclídea la BMU es la neurona más cercana a la muestra presentada en el espacio representado por todos los datos de entrada. La norma Euclídea de un vector x se define como:

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Dónde:

x_i: corresponde al valor de la componente i del vector x.

N: corresponde a la dimensión del vector x.

Por lo tanto, la distancia Euclídea en términos de la diferencia de la norma Euclídea entre dos vectores se define como:

$$d_E(x.y) = \|x - y\|$$

Dónde:

X: corresponde al vector x.

Y: corresponde al vector y.

La BMU, usualmente denotada con m_c, es el patrón que más se parece al vector de entrada x.

Se define formalmente como la neurona para la cual

$$\|x - m_c\| = \min_i \{\|x - m_i\|\}$$

Dónde:

X: corresponde al vector de entrada x.

m_c: corresponde al vector que representa la BMU.

i: corresponde a la neurona i.

m_i: corresponde al vector que representa la neurona mi.

Luego de encontrar la BMU, se actualizan todas las neuronas del SOM. Durante el procedimiento de actualización, la BMU se actualiza para acercarse aún más al vector de entrada. Los vecinos topológicos de la BMU también se actualizan de manera similar utilizando una tasa de aprendizaje de menor valor. Este procedimiento acerca a la BMU y a sus vecinos topológicos hacia la muestra ingresada.

El esfuerzo computacional consiste en encontrar una BMU entre todas las neuronas y actualizar cada uno de los patrones en el vecindario de la unidad ganadora. Si el vecindario es grande, entonces más patrones deberán ser actualizados; este es el caso que se presenta en el comienzo del entrenamiento, donde se recomienda utilizar vecindarios grandes. En el caso de redes con muchas neuronas, gran parte del tiempo se utiliza buscando a la ganadora. Obviamente que dependiendo del diseño del software utilizado y el hardware estas consideraciones serán más o menos significativas.

A través del procedimiento de actualización descrito, la red forma una red elástica que durante el aprendizaje cae en una nube formada por los datos de entrada. Los patrones tienden a posicionarse allí donde los datos son densos, mientras que se tiende a tener pocos patrones donde los datos de entrada están más dispersos. Por lo tanto, la red tiende a aproximar la función de densidad de probabilidad de los datos de entrada [Kohonen, 1995].

La regla de actualización del SOM para una unidad mi, es la siguiente:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)]$$

Dónde:

t: representa un estado en el tiempo.

Por lo tanto, y como se mencionó anteriormente, este es un proceso de entrenamiento a través del tiempo. El vector de entrada x(t) es tomado en instante

t para ser procesado, h_{ci} es una función de vecindario alrededor de la unidad ganadora m_c decreciente en el tiempo.

La función de vecindario que incluye la tasa de aprendizaje α(t) determina la forma en que serán actualizadas las neuronas vecinas. La misma se puede escribir como:

$$h_{ci}(t) = \alpha(t) e^{\frac{\|r_i - r_c\|^2}{2\sigma(t)^2}}$$

En el caso de una función de vecindario Gaussiana alrededor de la neurona m_c.

Se pueden utilizar otras funciones de vecindario como la función que se presenta en la figura 2.7. La única restricción es que sea decreciente alrededor de la neurona m_c. Por lo tanto, también podría ser constante alrededor de la neurona ganadora.

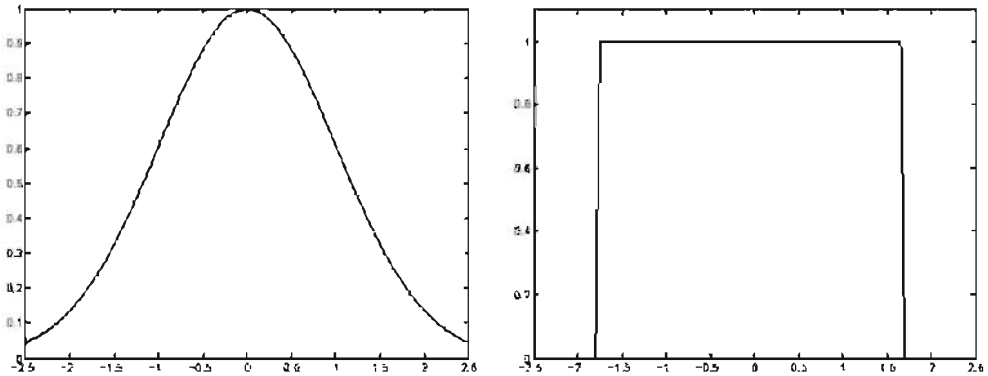


Figura 2.7: Funciones de vecindario

Fuente: [Kohonen, 1995].

En la figura 2.7 se pueden observar dos funciones de vecindario: **(a)** función Gaussiana, **(b)** función constante.

La tasa de aprendizaje utilizada en la función vecindario es una función decreciente en el tiempo. Dos formas comúnmente usadas son la función lineal y la inversamente proporcional al tiempo t.

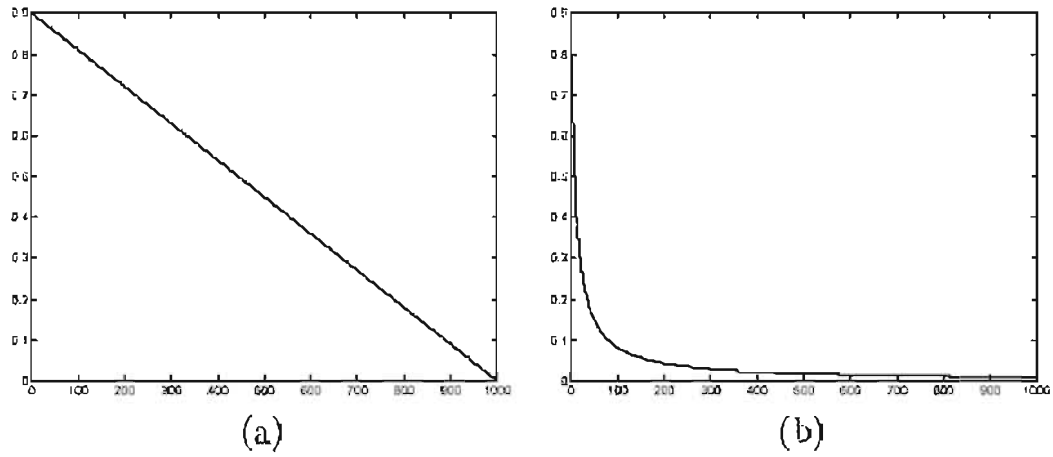


Figura 2.8: Tasas de aprendizaje
Fuente: [Kohonen, 1995].

En la figura 2.8 se pueden observar tipos de tasas de aprendizaje: **(a)** la función lineal decrece a cero linealmente durante el aprendizaje, **(b)** la función inversamente proporcional decrece rápidamente desde su valor inicial.

Los valores de la tasa de aprendizaje α se definen de la siguiente manera:

$\alpha(t) = \alpha(0)\left(1 - t/r\right)$, para el caso de la función inversa y

$\alpha(t) = C\alpha(0)/(C + t)$, para el caso de la función lineal donde C se puede definir como r/100 y r corresponde a la cantidad total de vectores muestra utilizados en el entrenamiento.

Se debe determinar el valor inicial de α, que define el valor inicial de la tasa de aprendizaje. Usualmente, cuando se utiliza una función inversa el valor inicial puede ser mayor que en el caso lineal. El aprendizaje se realiza usualmente en dos fases:

- En la primera vuelta se utilizan valores relativamente altos de α (desde 0,3 a 0,99).
- En la segunda vuelta se utilizan valores más pequeños. Esto corresponde a adaptaciones que se van haciendo hasta que la red funciona correctamente [Kohonen, 1995].

La elección de los valores iniciales de α y la forma en que estos van variando pueden variar sensiblemente los resultados obtenidos.

2.6.1.4. VISUALIZACION

El SOM es una aproximación de la función de densidad de probabilidad de los datos de entrada [Kohonen, 1995] y puede representarse de una manera visual [Livarinen, Kohonen, Kangas & Kaki, 1994].

La representación U-Matrix (unified distance Matrix) del SOM visualiza la distancia entre neuronas adyacentes [Kohonen, 1994]. La misma se calcula y se presenta con diferentes colores entre los nodos adyacentes. Un color oscuro entre neuronas corresponde a una distancia grande que representa un espacio importante entre los valores de los patrones en el espacio de entrada. Un color claro, en cambio, significa que los patrones están cerca unos de otros. Las áreas claras pueden pensarse como "clases" y las oscuras como "separadores". Esta puede ser una representación muy útil de los datos de entrada sin tener información a priori sobre las clases.

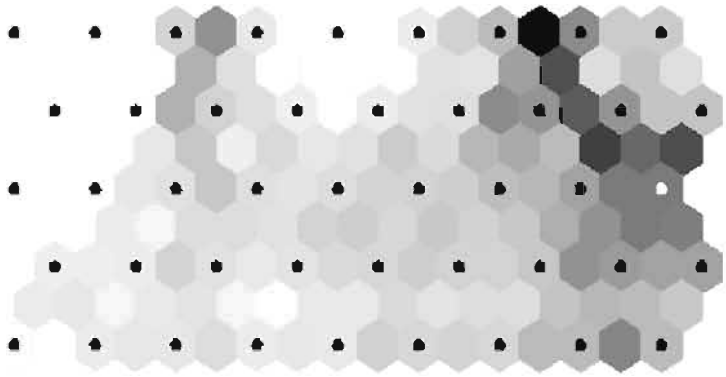


Figura 2.9: U-MATRIX
Fuente: [Kohonen, 1994].

En la figura 2.9 podemos observar las neuronas indicadas por un punto negro. La representación revela que existe una clase separada en la esquina superior

derecha de la red. Las clases están separadas por una zona negra. Este resultado se logra con aprendizaje no supervisado, es decir, sin intervención humana. Enseñar a un SOM y representarla con la U-Matrix ofrece una forma rápida de analizar la distribución de los datos.

2.6.1.5. VALIDACION

Se pueden crear la cantidad de modelos que se quiera, pero antes de utilizar alguno de ellos, deben ser validados. La validación significa que el modelo debe ser probado para asegurar que devuelve valores razonables y certeros. La misma debe realizarse usando un conjunto independiente de datos; este es similar al utilizado para el entrenamiento pero no parte de él; puede verse a este conjunto de prueba como un caso representativo del caso general.

2.6.2. APLICACIONES

Se ha demostrado que los SOM son muy útiles en aplicaciones técnicas. En la industria, se ha utilizado, por ejemplo, en monitoreo de procesos y máquinas [Alander & Frisk, 1991; Cumming, 1993; Alhoniemi, 1995], identificación de fallas [Vapola, Simula & Kohonen, 1994] y control de robots [Ritter, Martinetz & Schulten, 1992].

La capacidad de dividir el espacio en clases y patrones representativos lo hace muy poderoso también para la clasificación y segmentación de los datos; en el caso de estudio de este trabajo se presentan millones de llamadas y las redes SOM las clasifican y construyen patrones representativos del espacio total de las mismas [Grosser, G. Martínez, Sicre, Perichinsky, Serevetto & Britos, 2003].

2.6.3. PREDICCION DE CAMPOS ESTOCASTICOS GENERADOS POR REDES SOM

Una vez que la red neuronal ha determinado los patrones que representan el espacio de los datos de entrada, la red debe ponerse operativa. En el caso que la red funcione como clasificador, deberá determinar si el dato de entrada pertenece

a un patrón u otro sabiendo que aquella neurona que se encuentre a la mínima distancia será la que más probabilidad tenga de representar a dicho dato. Sin embargo, sería erróneo pensar que el dato ingresado corresponde cien por ciento al patrón más cercano, debido a que el espacio total se ha representado basado en datos experimentales también.

Por lo tanto sería más certero asignar una cierta probabilidad al dato ingresado que pertenezca a cada uno de los patrones. Para ello Grabec y Mandelj introdujeron el concepto de predicción de un campo basado en los patrones que lo representan al ingresar un nuevo dato [Grabec & Mandelj, 1998]. Grabec y Mandelj utilizan en su trabajo la función básica que puede determinar que tan "parecido" es un dato de entrada X a cada uno de los patrones de la red:

$$v_i = \frac{e^{-\|X-Q_i\|}}{\sum_{j=1}^{N} e^{-\|X-Q_j\|}}$$

Esta función corresponde a la medida normalizada de similitud entre los patrones Qi y el vector de datos de entrada X. Es decir que representa cuán lejos está el dato X de cada uno de los patrones generados y al ser $\sum_{i=1}^{N} v_i = 1$, puede ser visto como la probabilidad que la llamada X se parezca al patrón Q_i.

La ventaja de esta ecuación es que se define un nuevo patrón v, basado en los patrones existentes Q resultantes del entrenamiento de la red y luego puede usarse para corregir los patrones existentes convirtiéndose en un aprendizaje adaptativo ya que incorpora nueva información para representar el campo.

2.7. ANALISIS DE LA INFORMACION PARA LA DETECCION DE FRAUDE

La selección de información que debe ser analizada y luego procesada es la base de un buen sistema de detección de fraude [ASPeCT, 1997]. Una vez que se definieron los escenarios posibles de fraude (ver Clasificación de Tipos de Fraude), se identifican los indicadores típicos para detectarlos. Estos indicadores pueden ser clasificados en dos grupos diferentes [Moreau & Preneel, 1997]:

• Por tipo:

- *Indicadores de uso*: basados en la forma que se usa un teléfono celular.

- *Indicadores de movilidad*: basados en la información referente a la ubicación del teléfono celular.

- *Indicadores deductivos*, tales como solapamiento de llamadas y "velocity checks".

El solapamiento consiste en detectar dos llamadas realizadas en un mismo lapso de tiempo por el mismo teléfono, lo que seguramente resulta de una clonación. Los "velocity checks" también son indicadores de una posible clonación ya que detectan dos llamadas realizadas por el mismo teléfono en dos lugares alejados con horarios muy parecidos.

• Por uso:

- *Indicadores primarios*: son aquellos que por sí solos pueden ser empleados en la detección de fraude. Ejemplo: total de minutos de llamadas internacionales realizadas.

- *Indicadores secundarios*: son aquellos que proveen información muy útil, pero no son suficientes para detectar fraude por sí solos. Ejemplo: frecuencia de llamadas a un determinado destino.

- *Indicadores terciarios*: proveen información adicional que combinada con los indicadores anteriores pueden ser muy útiles. Ejemplo: duración promedio de las llamadas que realiza un determinado usuario.

Toda la información necesaria para medir los diferentes indicadores se encuentra en los CDR's, pero los fraudes más sofisticados no son detectables utilizando un único CDR. Para detectar tales fraudes es necesario investigar el consumo absoluto del usuario (análisis absoluto) y también los cambios en el comportamiento del mismo (análisis diferencial). La información del comportamiento de un usuario se analiza en dos períodos de tiempo o "ventanas" [Burge & Taylor, 1997]: una ventana referida al periodo reciente (CUP) y otra al periodo histórico (UPH); estos perfiles contienen información condensada en lugar de todo el detalle de los CDR's. Estos indicadores son utilizados por las

herramientas que se han desarrollado hasta el momento, cada una con un enfoque diferente, con sus ventajas y desventajas; cada una de ellas construye, codifica y procesa de manera diferente los perfiles CUP y UPH.

2.8. ENFOQUES PARA LA DETECCION DE FRAUDE

En esta sección se presentarán otras soluciones que se han desarrollado en el marco del uso de la inteligencia artificial para la detección de fraude.

2.8.1. ENFOQUE BASADO EN REGLAS

Este enfoque utiliza métodos automáticos de construcción y clasificación de perfiles de usuario con el propósito de encontrar fraude utilizando algunas técnicas de data mining que permiten construir las correspondientes reglas [Fawcett & Provost, 1997]. Específicamente se usan programas de aprendizajes de reglas para descubrir indicadores de fraude de una gran base de datos de clientes y sus correspondientes llamadas. Estos indicadores son utilizados luego para crear monitores, que clasifican el comportamiento legítimo y también las anomalías. Finalmente, la salidas de los monitores se usan como información para un sistema que aprende a combinar la evidencia para generar alarmas altamente confiables [Fawcett & Provost, 1997]. Este sistema se pensó para poder detectar, especialmente, fraude de clonación. Este fraude es un ejemplo de *fraude de súper imposición*, en el cual el uso fraudulento se agrega (se súper impone) al uso legítimo de la cuenta.

2.8.1.1. NATURALEZA ADAPTATIVA DE LA SOLUCION

Para poder construir los perfiles de usuario y que luego se pueda detectar fraude es necesario que los analistas ajusten los parámetros o ingresen valores específicos de umbrales que puedan emitir alarmas cuando son superados. Pero si estas reglas son estáticas o deben ser determinadas manualmente, esto resulta totalmente improductivo; además, los tipos de fraude evolucionan constantemente y por lo tanto son dinámicos [Sundaram, 1996]. Debido a esta realidad, es

necesario que el sistema de fraude se adapte fácilmente a las nuevas condiciones que se presentan constantemente. Utilizando técnicas de minería de datos es posible conseguir la adaptabilidad necesaria.

2.8.1.2. MODELO DE LA SOLUCION POR REGLAS

A continuación se presenta en la figura 2.10 un gráfico que describe el enfoque por reglas que han implementado Fawcett y Provost

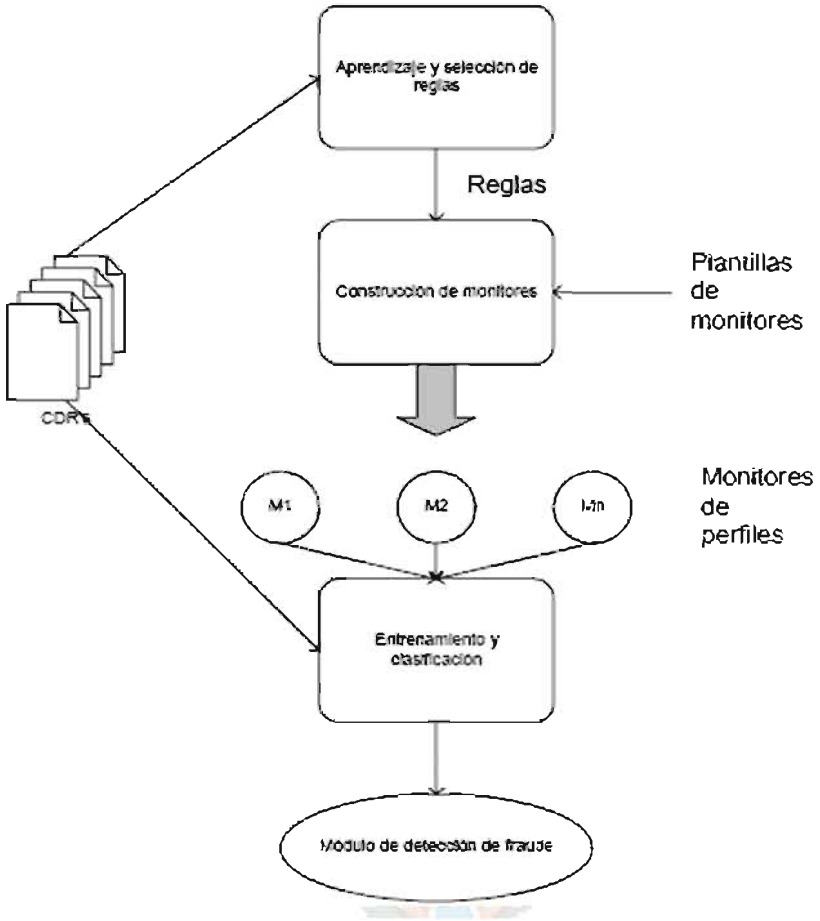


Figura 2.10: Enfoque basado en reglas

Fuente: [Fawcett y Provost].

En este enfoque, el sistema aprende primero las reglas que le servirán como indicadores de fraude. Luego utiliza estas reglas, a través de unas plantillas para crear los perfiles de monitores $(M_1 a M_n)$. Estos monitores clasifican el

comportamiento típico de cada usuario con respecto a una de las reglas definidas, que en realidad significa cuán lejos esta cada uno los usuarios de su comportamiento usual. Finalmente, el sistema aprende a ponderar las salidas de los monitores para maximizar la efectividad del detector de fraude. Los monitores reciben información de un día de llamadas de un determinado usuario, y cada uno de ellos genera un número indicando que tan diferente de lo usual parece el comportamiento del usuario en cuestión. Las salidas de todos los monitores son utilizadas para que el módulo de detección de fraude las combine y determine si la actividad es fraudulenta. De ser así, genera una alarma.

La primer etapa del aprendizaje de reglas se realiza en dos pasos: 1) las reglas se generan inicialmente localmente basadas en diferencias entre usos normales y usos fraudulentos de cada usuario. 2) luego son combinadas y se seleccionan si las mismas son válidas para un gran número de usuarios. Esto lo debe realizar analizando todas las reglas que se generaron para cada uno de los usuarios. Las reglas que se aprenden caracterizan cambios que ocurren comúnmente cuando se clona un teléfono celular. Pero como ya se ha citado, estas reglas no son universales. Por lo tanto el sistema debe saber cuándo aplicar las reglas a los usuarios que se están analizando y cuándo no hacerlo; esto se logra convirtiendo las reglas en monitores de perfiles [Davis & Goyal, 1993]. Los monitores de perfiles son creados por el constructor de monitores, que utiliza una serie de plantillas; las mismas se instancia por un conjunto de reglas de condición; con estas reglas y estas plantillas se genera un monitor.

En la siguiente etapa, el sistema aprende como combinar la evidencia de cada uno de los monitores generados en el paso previo; este módulo tiene como datos de entrada el consumo del usuario y el resultado de los monitores. Con toda la información recolectada, se le puede "enseñar" al sistema casos de fraude con determinado consumo y determinados valores de los monitores para que produzca una alarma cuando encuentra casos como los aprendidos; y también es necesario presentarle a esta red casos de uso normales (no fraudulentos) para que la misma identifique comportamientos dentro de los parámetros esperados.

2.8.1.3. LIMITACIONES DEL ENFOQUE POR REGLAS

Este enfoque tiene una gran ventaja en su capacidad constante de aprender diferentes escenarios de fraude por clonación basado en la información de cada uno de los usuarios y no generalizando reglas para todos ellos. Sin embrago, según lo que se describió, esta es una solución bastante compleja que requiere mucho procesamiento y una cantidad muy grande de información previa para que pueda comenzar a funcionar. Esta información incluye todo el consumo de los usuarios, (por lo menos de un día) y lo que hace que sea más difícil de implementar, una serie de casos fraudulentos para que los monitores se puedan construir con un grado de certeza tal que sirvan luego para la detección del uso fraudulento. Además, es una herramienta que se enfoca principalmente en el fraude por clonación, dejando de lado otros tipos de fraude tan importantes como éste.

Otro punto importante se refiere a la necesidad de un hardware acorde a las necesidades de procesamiento y almacenamiento de información que requiere, debido a que la combinación de datos de cada usuario produce reglas y a su vez las mismas luego deben ser agrupadas y analizadas para poder generar monitores más abarcativos. El procesamiento no finaliza aquí, ya que luego debe aprender cuáles casos son fraudulentos y cuáles no; esto implica un período muy importante de entrenamiento para el sistema en el cual el mismo no puede utilizarse para comenzar a trabajar, hasta tanto no tenga la cantidad suficiente de casos para analizar.

2.8.2. ENFOQUE BASADO EN REDES NEURONALES

Las redes neuronales usualmente proveen las mejores soluciones en situaciones donde es difícil establecer reglas definidas y rápidas y en las cuales los datos a analizar son complejos [Seymour, 2000]. Mientras más complejos son los datos, mayor es la ventaja de utilizar redes neuronales. También debido a su naturaleza aritmética, las redes neuronales son buenas procesando grandes volúmenes de información [Seymour, 2000].

Las redes neuronales son particularmente aptas para construir sistemas adaptativos, es decir, que aprenden de la experiencia [Hilera González & Martínez Hernando, 2000]. La habilidad de responder a los cambios de comportamiento a lo largo del tiempo y procesar grandes volúmenes de información, hacen que la detección de fraude sea una aplicación ideal para implementar con redes neuronales [Moreau & Vandervalle, 1997; ASPeCT, 1997]. En este ámbito, el comportamiento de los usuarios está siempre cambiando.

2.8.2.1. MODELO UTILIZANDO REDES NEURONALES SUPERVISADAS

El motor de detección de fraude en esta arquitectura asocia a cada usuario (IMSI), un CUP y un UPH. Aquí también se utiliza un CUR (Current User Record) que acumula información sobre los CDR's de un determinado lapso de tiempo, por ejemplo 1 día [ASPeCT, 1997]. Una vez que el CUR tiene la información necesaria de las llamadas de un día, se actualiza el CUP a través de la siguiente ecuación:

$$CUP_{i-1} = \alpha CUP_i - (1-\alpha)CUR$$

Dónde:

α: Es la tasa de adaptabilidad aplicada cuando el CUR se incorpora al CUP.

CUP_i: Estado del CUP en el instante i.

Esta técnica evita tener que almacenar todos CDR's del correspondiente usuario, almacenando solamente en el CUP una proporción de la información del CUR y quitando parte de la información más vieja del mismo a través del factor de adaptabilidad α.

De la misma manera, luego se actualiza el UPH con el CUP, obteniendo en dicho perfil información sobre el consumo histórico del usuario.

La información que contienen el CUP y el UPH es la siguiente:

- Media de la duración de las llamadas nacionales.
- Media de la duración de las llamadas internacionales.
- Varianza de la duración de las llamadas nacionales.
- Varianza de la duración de las llamadas internacionales.

- Tiempo promedio (media) entre dos llamadas nacionales.
- Tiempo promedio (media) entre dos llamadas internacionales.
- Varianza del tiempo entre dos llamadas nacionales.
- Varianza del tiempo entre dos llamadas internacionales.

La red neuronal supervisada, un Perceptrón multicapa [Hilera González & Martínez Hernando, 2000] es entrenada con CUR's, CUP's y UPH's de usuarios que hayan cometido fraude y usuarios normales para que la misma pueda clasificarlos luego en fraudulentos o no fraudulentos.

A continuación se presenta en la figura 2.10 un gráfico donde se esquematiza el funcionamiento del sistema en la etapa de entrenamiento:

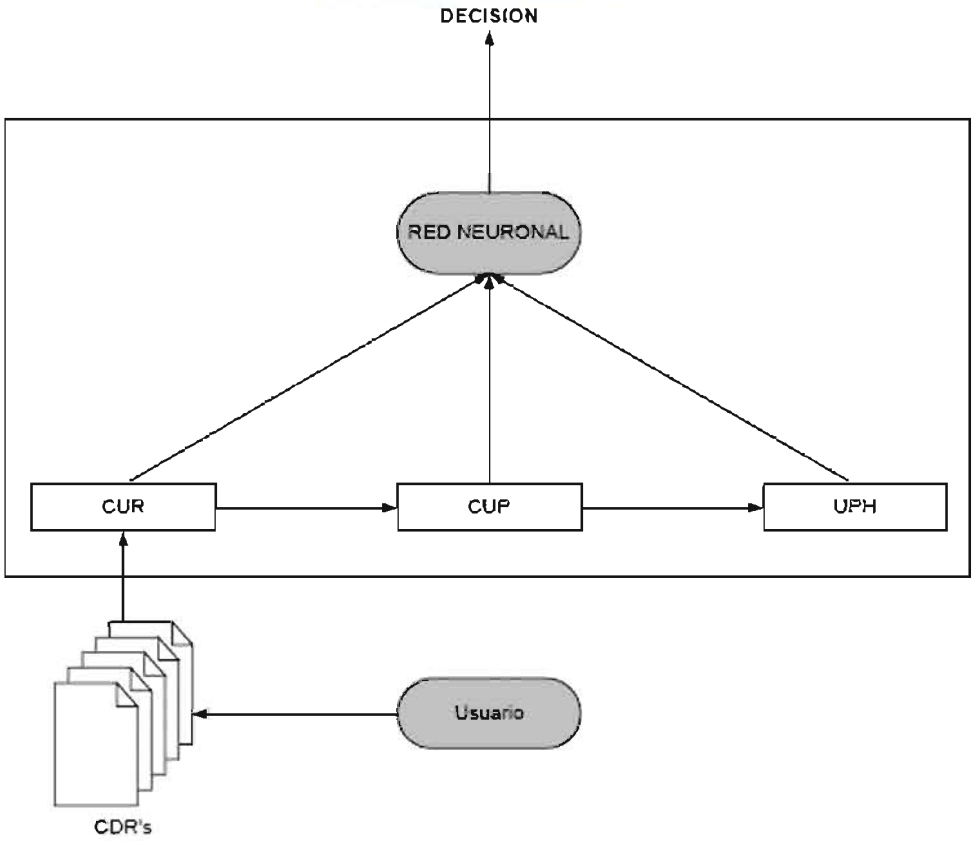


Figura 2.11: Enfoque basado en redes neuronales supervisadas

Fuente: [ASPeCT, 1997].

El usuario realiza las llamadas y se generan los CDR's; luego se construye el CUR con la información de los mismos y se adapta el CUP y el UPH; con esta información se entrena a la red neuronal para que devuelva los resultados esperados para dicha combinación de CUR, CUP y UPH. En la etapa de régimen permanente (funcionamiento del motor de detección de fraude), a medida que ingresan los CDR's se van actualizando el CUR y CUP del usuario; luego ingresan para ser analizados junto con el UPH y si la red no lanza ninguna alarma, se actualiza el UPH. La información de los CUP y UPH se almacenan en una base de datos para ser recuperadas cada vez que ingresan al sistema CDR's con información de los usuarios correspondientes.

2.8.2.2. LIMITACIONES DEL ENFOQUE BASADO EN REDES SUPERVISADAS

Este enfoque presenta una gran limitación en la necesidad de tener que ser constantemente entrenado con nuevos casos de fraude que van apareciendo debido a que tiene dos etapas definidas: una de entrenamiento y una de funcionamiento. En el caso de aparecer nuevos tipos de fraude será necesario sacar de línea el sistema para que incorpore los nuevos casos; es decir que no aprovecha el potencial de la naturaleza adaptativa del problema. Esta necesidad de tener casos de fraude a priori también obliga a quienes lo quieran implementar que posean información previa de casos existentes, cuando muchas veces no es posible obtenerla o no se conoce.

2.8.2.3. MODELO UTILIZANDO REDES NEURONALES NO SUPERVISADAS

La solución que se propone en este trabajo utiliza redes neuronales no supervisadas para construir los perfiles de usuario [Burge & Taylor, 1997]; en nuestro caso se utilizan redes SOM que como resultado logran clasificar las millones de llamadas que se procesan en una cantidad determinadas de prototipos que representan todo el espacio de las mismas. La frecuencia con la cual un usuario realiza llamadas de cada prototipo corresponde a la representación de los

perfiles CUP y UPH. Una vez que ambos se actualizan, se comparan y se decide si la diferencia entre el consumo reciente y el histórico es lo suficientemente grande como para emitir una alarma.

2.9. MARCO LEGAL

LEGISLACION BOLIVIANA EN MATERIA DE DELITOS INFORMATICOS

En el tema de legislación de los delitos informáticos, en Bolivia estamos un poco atrasados, solo tenemos 2 delitos de los 15 delitos establecidos por la ONU-OMPI. Se tiene manipulación informática y acceso indebido en el Código Penal. Lo lamentable es que el acceso indebido no tiene cárcel. Sin embargo de estas debilidades de nuestra legislación, cabe mencionar que estos delitos cuando se cometen, casi siempre van de la mano de otros delitos antiguos y ya tipificados en el Código Penal. Por ejemplo se realiza la manipulación de los datos (mediante phishing se captura datos de un cliente de un banco) en el proceso de entrada de datos, y esta información se utiliza para sacar dinero de esa cuenta, puede ser vía caja, transferencia, ATM, entonces el delito informático (digital) se concreta en algo físico ROBO. Así como el caso comentado, estos dos delitos (manipulación y acceso indebido) se pueden vincular con otros, como son el robo, hurto, uso de instrumento falsificado, abuso de confianza y otros.

Aquí se tienen una trascripción de los dos Artículos dentro el Código Penal.

CAPITULO XI

DELITOS INFORMATICOS

Articulo 363.-bis(MANIPULACION INFORMATICA).El que con la intención de obtener un beneficio indebido para sí o un tercero, manipule un procesamiento o transferencia de datos informáticos que conduzca a un resultado incorrecto o evite un proceso tal cuyo resultado habría sido correcto, ocasionando de esta manera una transferencia patrimonial en perjuicio de tercero, será sancionado con reclusión de uno a cinco años y con multa de sesenta a doscientos días.

Articulo 363.-ter(ALTERACION, ACCESO Y USO INDEBIDO DE DATOS INFORMATICOS). El que sin estar autorizado se apodere, acceda, utilice, modifique, suprima o inutilice, datos almacenados en una computadora o en cualquier soporte informático, ocasionando perjuicio al titular de la información, será sancionado con prestación de trabajo hasta un año o multa hasta doscientos días."

Existen también dentro de la NUEVA CONSTITUCION POLITICA DEL ESTADO, dos Artículos referentes al tema de Delitos Informáticos, y hacen referencia a lo siguiente:

SECCIÓN III

ACCIÓN DE PROTECCIÓN DE PRIVACIDAD

Artículo 133

I. Toda persona individual o colectiva que crea estar indebida o ilegalmente impedida de conocer, objetar u obtener la eliminación o rectificación de los datos registrados por cualquier medio físico, electrónico, magnético o informático, en archivos o bancos de datos públicos o privados, o que afecten a su derecho fundamental a la intimidad y privacidad personal y familiar, a su propia imagen, honra y reputación, podrá interponer la Acción de Protección de Privacidad.
II. La Acción de Protección de Privacidad no procederá para levantar el secreto en materia de prensa.

Artículo 134

I. La Acción de Protección de Privacidad tendrá lugar de acuerdo con el procedimiento previsto para la acción de Amparo Constitucional.
II. Si el tribunal o juez competente declara procedente la acción, ordenará la revelación, eliminación o rectificación de los datos cuyo registro fue impugnado.
III. La decisión se elevará en revisión de oficio ante el Tribunal Constitucional Plurinacional, en el plazo de las veinticuatro horas siguientes a la emisión del fallo, sin que por ello se suspenda su ejecución.

IV. La decisión final que conceda la Acción de Protección de Privacidad será ejecutada inmediatamente y sin observación. En caso de resistencia se procederá de acuerdo a lo señalado en la Acción de Libertad. La autoridad judicial que no proceda conforme a lo dispuesto por este artículo, quedará sujeta a las sanciones previstas por la ley.

CAPITULO III
MARCO APLICATIVO

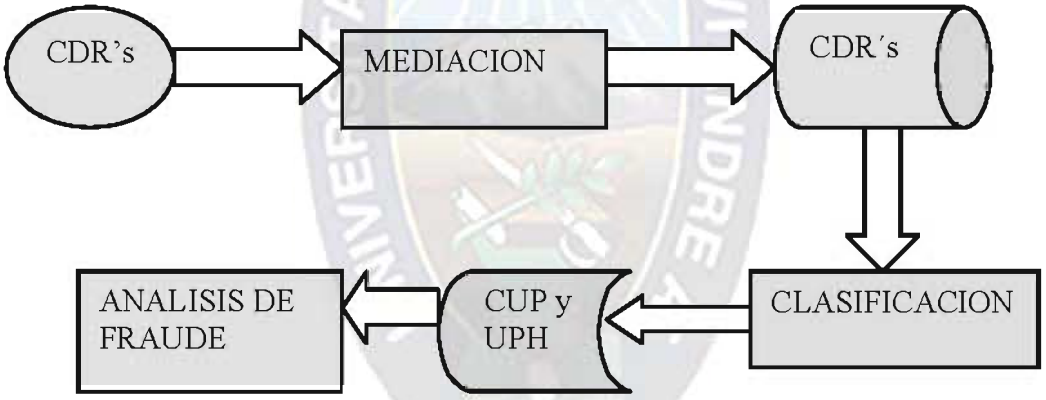
CAPITULO III

MARCO APLICATIVO

3.1. INTRODUCCION

El desarrollo de una red neuronal artificial exige una metodología a seguir de manera que los resultado esperados sean los deseados y se pueda demostrar la hipótesis establecida, esta metodología se deberá adecuar al problema que se está tratando de resolver como es la detección de fraude en telefonías celulares. Para este propósito se utilizo el algoritmo SOM.

MODELO DEL PROBLEMA A RESOLVER



Fuente: [Elaboración propia].

Los CDR's ingresan y deben pasar primero por un proceso de simplificación y traducción a un formato de registro en el cual solamente se almacenen los datos necesarios, los cuales deberán tener un formato conocido para que el siguiente proceso los tome; el mismo se denomina mediación. El proceso de clasificación es el responsable de tomar cada una de las llamadas, con la ayuda de las redes neuronales y generar (o actualizar) los perfiles de usuario CUP y UPH. Una vez que estos perfiles se han actualizado con la información de las llamadas recibidas, se realiza el proceso de análisis de fraude por comparación de ambos perfiles.

3.2. ANALISIS DE LA INFORMACION PARA DETECCION DE FRAUDE

La selección de información que debe ser analizada y luego procesada es la base de un buen sistema de detección de fraude. Una vez que se definieron los escenarios posibles de fraude (ver Clasificación de Tipos de Fraude), se identifican los indicadores típicos para detectarlos. Estos indicadores pueden ser clasificados en dos grupos diferentes:

• Por tipo:

- *Indicadores de uso*: basados en la forma que se usa un teléfono celular.
- *Indicadores de movilidad*: basados en la información referente a la ubicación del teléfono celular.
- *Indicadores deductivos*, tales como solapamiento de llamadas y "velocity checks".

El solapamiento consiste en detectar dos llamadas realizadas en un mismo lapso de tiempo por el mismo teléfono, lo que seguramente resulta de una clonación. Los "velocity checks" también son indicadores de una posible clonación ya que detectan dos llamadas realizadas por el mismo teléfono en dos lugares alejados con horarios muy parecidos.

• Por uso:

- *Indicadores primarios*: son aquellos que por sí solos pueden ser empleados en la detección de fraude. Ejemplo: total de minutos de llamadas internacionales realizadas.
- *Indicadores secundarios*: son aquellos que proveen información muy útil, pero no son suficientes para detectar fraude por sí solos. Ejemplo: frecuencia de llamadas a un determinado destino.
- *Indicadores terciarios*: proveen información adicional que combinada con los indicadores anteriores pueden ser muy útiles. Ejemplo: duración promedio de las llamadas que realiza un determinado usuario.

Toda la información necesaria para medir los diferentes indicadores se encuentra en los CDR's, pero los fraudes más sofisticados no son detectables utilizando un

único CDR. Para detectar tales fraudes es necesario investigar el consumo absoluto del usuario (análisis absoluto) y también los cambios en el comportamiento del mismo (análisis diferencial). La información del comportamiento de un usuario se analiza en dos períodos de tiempo o "ventanas": una ventana referida al periodo reciente (CUP) y otra al periodo histórico (UPH); estos perfiles contienen información condensada en lugar de todo el detalle de los CDR's.

Para poder comenzar a procesar los CDR's se debe crear un formato de registro (salida de la mediación) con la siguiente información: IMSI (identifica al usuario), fecha de la llamada, hora de la llamada, duración de la llamada y tipo de llamada clasificada en LOC (llamada local), NAC (llamada nacional) e INT (llamada internacional). Con esta información ya acotada a los datos necesarios, se pueden comenzar a resolver las siguientes y más importantes cuestiones utilizando como datos de entrada la salida de la mediación.

3.3. SOLUCION A LA CONSTRUCCION DE "PERFILES DE USUARIO"

La primera cuestión a resolver es determinar cómo construir los perfiles CUP y UPH; es decir, que se debe determinar los *patrones* que compondrán cada uno de estos perfiles. Los patrones deberán tener información del consumo del usuario, separando el consumo LOC, NAC e INT respectivamente. Una forma de construir estos patrones es utilizando redes neuronales para discretizar el espacio de todas las llamadas de los usuarios, generando un espacio de n patrones que representen el consumo de todos los usuarios y luego generando una distribución de frecuencias por cada usuario en la cuál se represente qué probabilidad de hacer llamadas de ese patrón tiene un usuario. En resumen, cuando se construya el perfil de usuario se estará representando la distribución de frecuencia en la cuál un determinado usuario realiza un tipo de llamada determinado, mostrando esta estructura de datos el *patrón de consumo* del mismo. Las redes neuronales, entre otras ventajas, tienen la capacidad de clasificar la información en determinados patrones [Hilera González & Martínez Hernando, 2000]; en especial, las redes

SOM (Self Organizing Map) pueden tomar esta información y construir estos patrones de manera *no supervisada* por criterios de semejanza. En nuestro caso, se pueden procesar todas las llamadas realizadas por todos los usuarios para que las redes, según la cantidad que hay de cada tipo genere los patrones que representen a todas ellas. Para evitar ruidos en los datos, se utilizan 3 redes neuronales que generen patrones para representar a las llamadas LOC, NAC e INT respectivamente; el perfil de usuario se construye utilizando todos los patrones generados por las 3 redes. Los datos que se utilizan para representar un patrón son la hora de la llamada y la duración de la misma; sabemos que si representamos en un eje cartesiano la hora de todas las llamadas y la duración correspondiente, obtendremos un rectángulo prácticamente lleno de puntos. La idea es obtener un gráfico en el que sólo aparezcan los puntos más representativos de todo el espacio en cuestión; esa es la tarea de las redes neuronales. Este diseño de 3 redes neuronales permite, no solamente detectar cambios de comportamiento sino que también representa de manera general el comportamiento de todos los usuarios de la compañía; es decir, que visualizando los patrones generados por cada una de las redes neuronales en un gráfico, podemos fácilmente obtener conclusiones de cómo se comportan en general los usuarios de la compañía y basado en ello, tomar decisiones del tipo comercial, agregando una funcionalidad más a la solución diseñada.

Una vez obtenidos los patrones que se utilizarán para representar los perfiles de usuario, es necesario comenzar a *llenar de información* a los mismos; el procedimiento consiste en tomar la llamada a analizar, y que la red neuronal determine a qué patrón se parece más la misma; una vez obtenida esta información, se debe adaptar el perfil de usuario CUP de manera que la distribución de frecuencia muestre que el usuario tiene ahora una probabilidad mayor de realizar este tipo de llamadas. Sabiendo que el perfil de usuario tiene *K* patrones que se componen de *L* patrones LOC, *N* patrones NAC e *I* patrones INT, podemos construir un perfil representativo de la llamada procesada y luego *adaptar* el perfil CUP con dicha llamada; si la llamada es LOC, los *N* patrones NAC y los *I* patrones INT tendrán una distribución de frecuencia igual a 0, y los *K*

patrones LOC tendrán una distribución de frecuencia dada por la ecuación de predicción de campos estocásticos en redes SOM:

$$v_i = \frac{e^{-|x-Q_i|}}{\sum_{j=1}^{L} e^{-|x-Q_j|}}$$

Dónde:

X: llamada a procesar

v_i : Probabilidad que la llamada X sea del patrón i.

Q_i : Patrón i generado por la red neuronal LOC.

Nótese que:

$$\sum_{j=1}^{L} v_j = 1.$$

Si la llamada fuese NAC, entonces se debe reemplazar L por *N* y la distribución de frecuencias LOC e INT serán 0; si la llamada fuese INT, entonces se debe reemplazar *L* por *I* y la distribución de frecuencias LOC e NAC serán 0,

Entonces, podemos definir el vector representativo de la llamada *V*, de dimensión *K* como:

$V_i = v_i$, con $1 \le i \le L$

$V_i = 0$, con $L+1 \le i \le K$, cuando la llamada es LOC.

$V_i = v_i$, con $L+1 \le i \le L+N$

$V_i = 0$, con $1 \le i \le L$ y $L+N \le i \le K$, cuando la llamada es NAC.

$V_i = v_i$, con $L+N+1 \le i \le K$

$V_i = 0$, con $1 \le i \le L+N$, cuando la llamada es INT.

Ahora que tenemos el vector *V*, podemos adaptar el vector *CUP* con la información de la llamada procesada:

$CUP_i = \alpha_{LOC} CUP_i - (1-\alpha_{LOC})V_i$, con $1 \le i \le K$, Cuando la llamada es LOC,

$CUP_i = \alpha_{NAT} CUP_i - (1-\alpha_{NAT})V_i$, con $1 \le i \le K$, Cuando la llamada es NAC,

$CUP_i = \alpha_{INT} CUP_i - (1-\alpha_{INT})V_i$, con $1 \le i \le K$, Cuando la llamada es INT, dónde

αLOC: Es la tasa de adaptabilidad aplicada cuando la llamada X se incorpora al CUP, si X corresponde a una llamada local.

αNAC: Es la tasa de adaptabilidad aplicada cuando la llamada X se incorpora al CUP, si X corresponde a una llamada nacional.

αINT : Es la tasa de adaptabilidad aplicada cuando la llamada X se incorpora al CUP, si X corresponde a una llamada internacional.

Una vez adaptado el perfil CUP, se compara con el perfil UPH y se determina si ha habido un cambio significativo de comportamiento (motor de detección de cambios de comportamiento); una vez realizada esta tarea, se adapta el UPH con la información del CUP solamente si la cantidad de llamadas necesarias para cambiar el patrón histórico se han procesado:

$$UPH_i = \beta UPH_i \div (1-\beta) CUP_i, \text{ con } 1 \le i \le K$$

Dónde:

β: Es la tasa de adaptabilidad aplicada cuando el CUP se incorpora al UPH.

3.4. SOLUCION A LA DETECCION DE CAMBIOS DE COMPORTAMIENTO

Para determinar si hubo o no cambios en el patrón de comportamiento, es necesario comparar los perfiles CUP y UPH y decidir si la diferencia entre los mismos es lo suficientemente grande como para lanzar una alarma. Debido a que el CUP y el UPH son dos vectores que representan distribuciones de frecuencia, se puede utilizar una distancia vectorial para comparar qué tan diferentes son. Para ello se puede utilizar la distancia Hellinger (H) cuyo valor indica la diferencia entre dos distribuciones de frecuencia. La distancia siempre será un valor entre cero y dos donde cero es para distribuciones iguales y dos representa ortogonalidad. El valor de H determinará qué tan diferentes deben ser las distribuciones de frecuencia CUP y UPH.

$$H = \sum_{i=1}^{K} \left(\sqrt{CUP_i} - \sqrt{UPH_i} \right)^2$$

Esta ecuación define la forma de detectar las diferencias entre el comportamiento reciente y el histórico.

3.5. SOLUCION A LAS CUESTIONES DE PERFORMANCE

La performance dependerá directamente del Hardware donde corra el sistema de detección de fraude y cambios de comportamiento. Desde el punto de vista del software se trabaja lo menos posible con bases de datos relacionales y se trata de hacer todo el procesamiento utilizando archivos planos de datos, con la mínima cantidad de escrituras y lecturas de disco. Es importante la compresión de los mismos ya que el espacio es otra restricción que se debe tener prevista. Por lo tanto, en la solución propuesta solo se trabaja con archivos planos y se almacena un archivo por usuario con la información de las distribuciones CUP y UPH, así como también la última llamada procesada y la cantidad total de llamadas procesadas.

3.6. METOLOGIA UTILIZADA

Los experimentos se dividieron en dos partes: la primera se enfocó en el entrenamiento de la red y la generación de los patrones para construir posteriormente los perfiles de usuario; la segunda prueba se enfocó en el análisis de las llamadas de los usuarios con alto consumo y el correspondiente análisis.

3.6.1. EXPERIMENTOS DE GENERACION DE PATRONES

Se construyeron 3 redes neuronales Self Organizing Map (SOM) para la generación de los patrones para las llamadas locales (LOC), NAC e INT respectivamente. Cada una de las redes fue entrenada con una cantidad de llamadas representativa del consumo de los usuarios de la empresa que los mismos realizaron durante unos días en todos los horarios. Las llamadas se presentaron a las redes de manera desordenada de manera que los patrones que se generaron no fueran solamente representativos de los horarios y duraciones de las últimas llamadas. El resultado de esta experiencia definió los patrones para construir los perfiles de los usuarios. Los patrones se componen de la hora de la llamada y la duración en minutos de la misma. Estos patrones lograron discretizar

el espacio compuesto por todos los tipos de llamada realizadas por cualquier usuario en una cantidad fija representativa del mismo.

3.6.2. EXPERIMENTOS DE CONSTRUCCION DE PERFILES Y DETECCION DE COMPORTAMIENTOS

Una vez obtenidos los patrones que definen el espacio de todas las llamadas, se realizaron las pruebas de construcción de los perfiles de usuario a través del desarrollo de una distribución de frecuencias de cada uno de los patrones para cada perfil (CUP y UPH). El proceso se basó en presentar las llamadas realizadas en un período de 3 meses por los usuarios reportados como "alto consumo". Con cada llamada se actualizaba el perfil CUP del usuario, se comparaba con el perfil UPH. Dependiendo del parámetro de frecuencia de actualización del perfil UPH (f), se actualizaba el UPH con el aporte del CUP según corresponda. Vale aclarar que el proceso de construcción y actualización se hizo desde la primera llamada del usuario, en cambio la comparación se realizó solamente luego que la cantidad de llamadas analizadas para el usuario pasara la cantidad mínima para construir un perfil (QL) con la suficiente información del usuario.

En el momento de ingresar la primera llamada de un usuario, se inicializaba a todos los patrones del CUP y UPH con la misma distribución de frecuencia, asumiendo que el usuario tenía la misma tendencia a realizar cualquier tipo de llamada a priori, sin información.

Esta experiencia se realizó dos veces: la primera actualizando el UPH con cada llamada debido a que la diferencia que se pudiera presentar entre los perfiles CUP y UPH era muy pequeña actualizando el perfil histórico con cada llamada, ya que el mismo tendía a ser igual al perfil actual. La segunda experiencia se realizó actualizando el UPH una vez por día para detectar diferencias importantes que puedan ser consideradas como cambios de comportamiento.

Los factores de adaptabilidad de llamadas en el CUP (αLOC, αNAT, αINT) y el factor de adaptabilidad de UPH (β) fueron determinados experimentalmente realizando varias iteraciones hasta observar que el agregado de información a los

perfiles CUP y UPH no implicaba perder todo lo aprendido anteriormente; estos valores fueron variados entre 0,6 y 0,9 hasta encontrar resultados satisfactorios.

3.7. PARAMETROS UTILIZADOS PARA LA GENERACION DE PATRONES

3.7.1. PARAMETROS INDEPENDIENTES

Los valores utilizados para la generación de los perfiles fueron los siguientes:

- *Dimensión de la red neuronal para clasificar llamadas locales* (NLxML) = 12x12
- *Dimensión de la red neuronal para clasificar llamadas nacionales* (NNxMN) = 8x8
- *Dimensión de la red neuronal para clasificar llamadas internac.* (NIxMI) = 6x6
- *Tasa de aprendizaje estática* (α) = 0,6
- *Distancia máxima de neurona "vecina" afectada* (DVMAX) = 10

3.7.2. PARAMETROS DEPENDIENTES

Los mismos definen la dimensión de los perfiles CUP y UPH:

- *Cantidad de patrones para clasificar las llamadas locales* (PL) = 144
- *Cantidad de patrones para clasificar las llamadas nac.* (PN) = 64
- *Cantidad de patrones para clasificar las llamadas internac.* (PI) = 36
- *Dimensión de los perfiles CUP y UPH* (K) = 244

3.8. PARAMETROS UTILIZADOS PARA LA CONSTRUCCION DE PERFILES Y DETECCION DE CAMBIOS DE COMPORTAMIENTOS

Los valores utilizados para la construcción de perfiles y detección de alarmas fueron los siguientes:

Experiencia 1:

- *Factor de adaptabilidad de llamadas locales en el CUP* (αLOC) = 0,8
- *Factor de adaptabilidad de llamadas locales en el CUP* (αNAC) = 0,8
- *Factor de adaptabilidad de llamadas locales en el CUP* (αINT) = 0,8
- *Factor de adaptabilidad de UPH* (β) = 0,9

- *Frecuencia de actualización del UPH* (f) = 1 llamada.

Experiencia 2:

- *Factor de adaptabilidad de llamadas locales en el CUP* (αLOC) = 0,8
- *Factor de adaptabilidad de llamadas locales en el CUP* (αNAC) = 0,9
- *Factor de adaptabilidad de llamadas locales en el CUP* (αINT) = 0,9
- *Factor de adaptabilidad de UPH* (β) = 0,6
- *Frecuencia de actualización del UPH* (f) = 1 día.

Valores comunes a ambas pruebas:

- *Cantidad mínima de llamadas antes de comparar perfiles* (QL) = 100 llamadas.

3.9. RESULTADOS

3.9.1. GENERACION DE PATRONES

En esta sección se presentan los resultados obtenidos luego del entrenamiento de las 3 redes neuronales; es decir, que los resultados muestran cada uno de los patrones que las redes "determinaron" como más representativos del espacio de todas las llamadas de todos los usuarios. Se presentan 3 gráficos (uno por cada red) en el que se muestra los patrones generados. En el eje X se muestra la hora de la llamada y en el eje Y la duración expresada en minutos. Cada uno de los puntos representados corresponde a un patrón elegido por la red como representativo de la muestra.

En el gráfico de la red neuronal local, se muestran 144 patrones; en el de la red NAC, 64 y en el de la red INT 36.

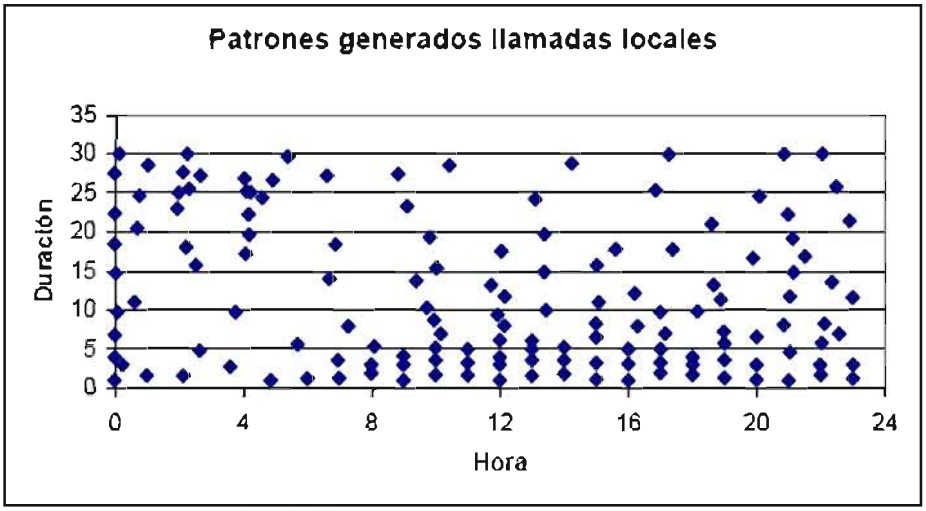


Gráfico 3.1: Patrones llamadas locales

Se observa en el gráfico 3.1 los 144 patrones generados luego del entrenamiento de la red neuronal de llamadas locales. A simple vista se puede notar que hay una concentración mayor de patrones en la banda horaria de las 8 hs. a las 20 hs y una duración entre 0 y 5 minutos. Esto denota que la mayoría de las llamadas locales realizadas por los clientes de esta empresa ocurren en estos horarios con los promedios de duración indicados.

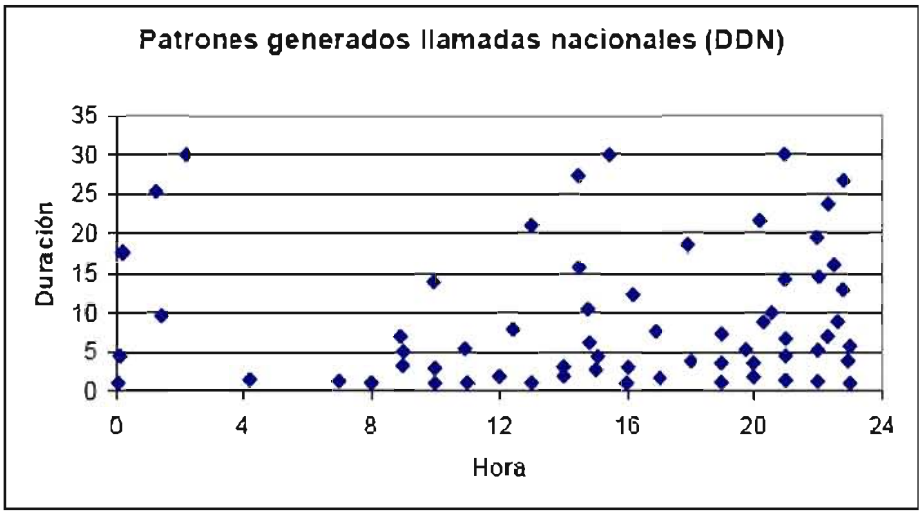


Gráfico 3.2: Patrones llamadas nacionales

Se observa en el gráfico 3.2 los 64 patrones generados luego del entrenamiento de la red neuronal de llamadas nacionales; aquí también se observa una

concentración de patrones, pero más desplazada hacia la banda horaria de las 15 a las 22 con duraciones que oscilan entre los 0 y 7 minutos; también se observa que prácticamente no hay patrones generados para la madrugada, con lo cual se puede concluir que la mayoría de los usuarios de la empresa analizada no realizan llamadas NAC en horas muy tempranas.

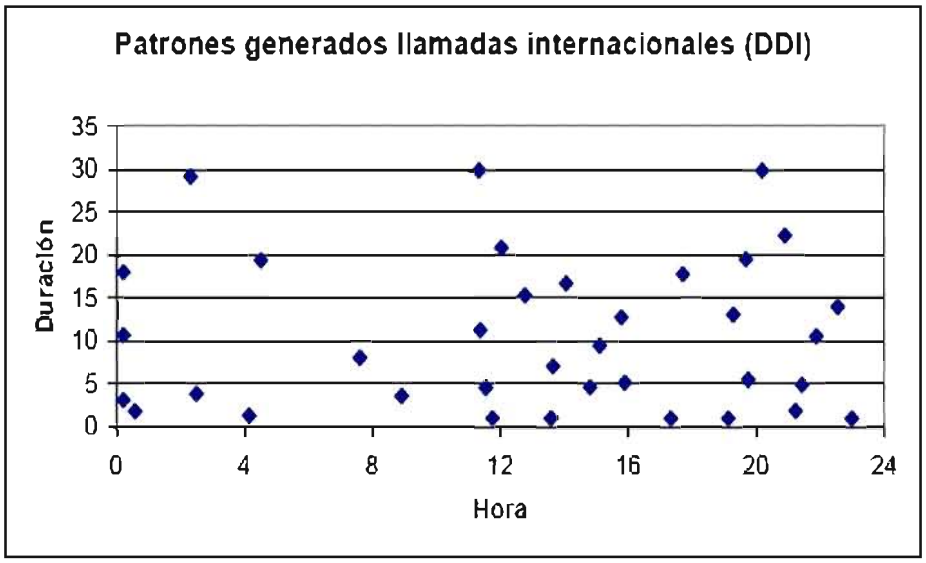


Gráfico 3.3: Patrones llamadas internacionales

Se observa en el gráfico 3.3 los 36 patrones generados luego del entrenamiento de la red neuronal de llamadas internacionales; aquí la distribución es un poco más aleatoria, pero la duración de las llamadas "elegidas" como patrones tiende a tener una duración mayor (entre 7 y 10 minutos).

3.9.2. CONSTRUCCION DE PERFILES Y DETECCION DE CAMBIOS DE COMPORTAMIENTOS

En esta sección se presentan los resultados obtenidos luego de la construcción de los perfiles. Se muestran los gráficos 3.4, 3.5, 3.6 y 3.7 con una descripción de los perfiles CUP y UPH de uno algunos casos. En el eje X se presentan los 244 patrones (144 LOC, 64 NAC y 36 INT) y en el eje Y la distribución de frecuencias de cada uno de los patrones para el usuario analizado. También se realizará un

análisis de la confiabilidad y veracidad de las mismas basadas en el detalle de llamadas de cada usuario.

Experiencia 1 (Actualización UPH con cada llamada, alta sensibilidad con bajo Umbral Hellinger):

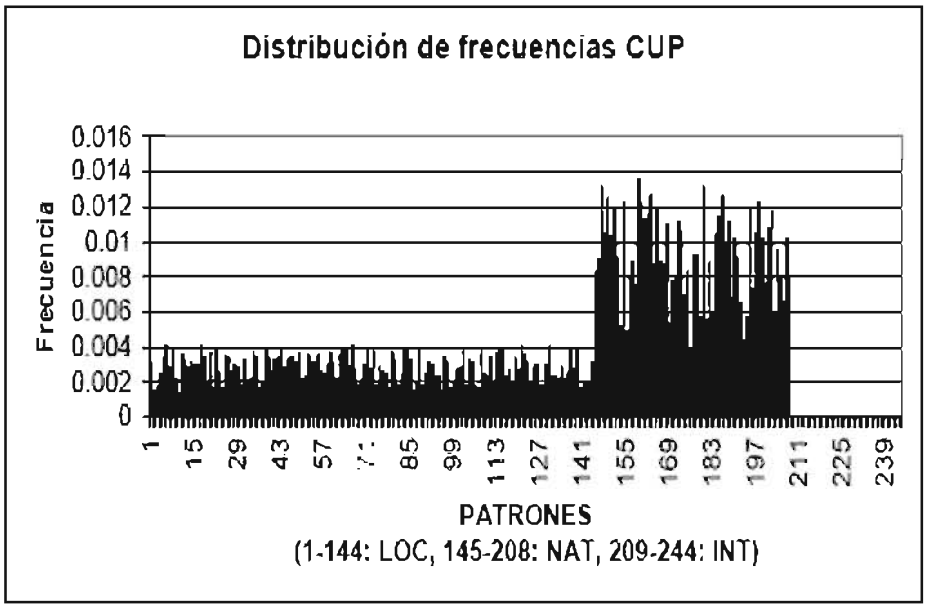


Gráfico 3.4: Distribución de frecuencias CUP experiencia 1

El gráfico 3.4 muestra el CUP de un usuario. Se puede observar en el mismo que la distribución de frecuencias indica una mayor tendencia a realizar llamadas NAC (patrones 145 a 208).

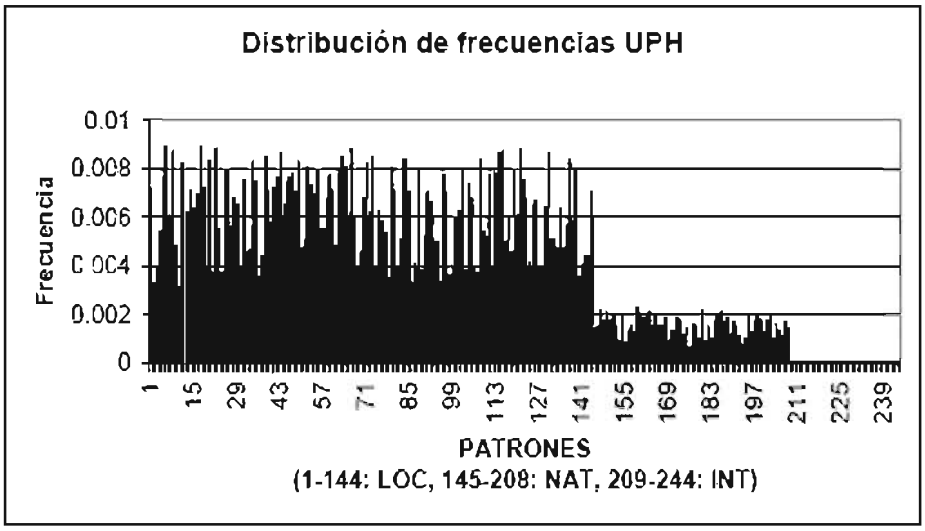


Gráfico 3.5: Distribución de frecuencias UPH experiencia 1

El gráfico 3.5 muestra el UPH del mismo usuario. Se puede observar en el mismo que la distribución de frecuencias indica una mayor tendencia a realizar llamadas locales (patrones 1 a 144).

Analizando el detalle de llamadas de este usuario desde fechas anteriores, se observa que el usuario hizo una llamada NAC por primera vez desde que se procesaron sus llamadas; es decir, que su patrón de comportamiento histórico no hacía creer que iba a realizar llamadas de este tipo. También muestran estos resultados que al haber realizado la experiencia con tan alta sensibilidad, una llamada diferente puede indicar un cambio de comportamiento. La mayoría de las mismas siguen el patrón del caso que se muestra en los gráficos 3.4 y 3.5 en el cual una llamada diferente al patrón normal de comportamiento alcanza para que el sistema defina al usuario como sospechoso.

Experiencia 2 (Actualización UPH una vez por día, moderada sensibilidad con Umbral Hellinger):

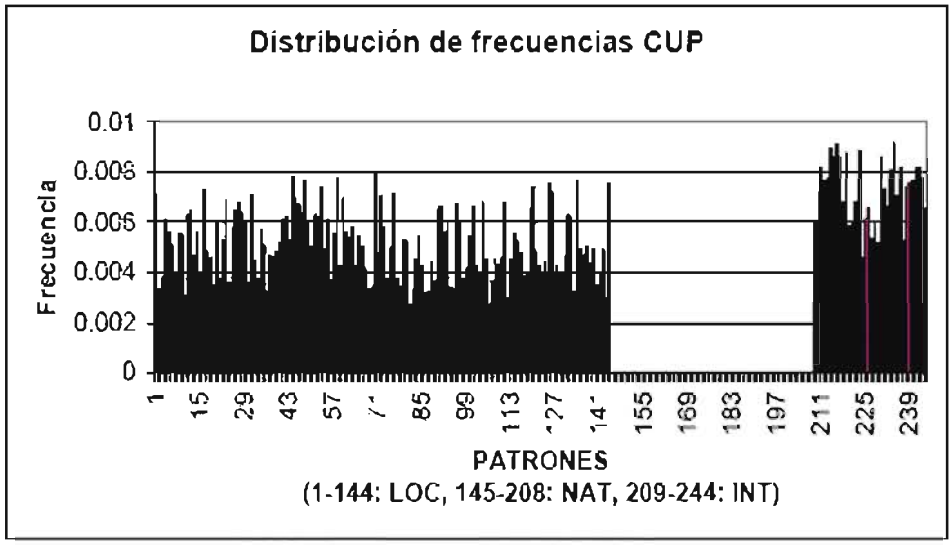


Gráfico 3.6: Distribución de frecuencias CUP experiencia 2

El gráfico 3.6 muestra el CUP de un usuario. Se puede observar en el mismo que la distribución de frecuencias indica una tendencia a realizar llamadas locales (patrones 1 a 144) e internacionales (patrones 209 a 244).

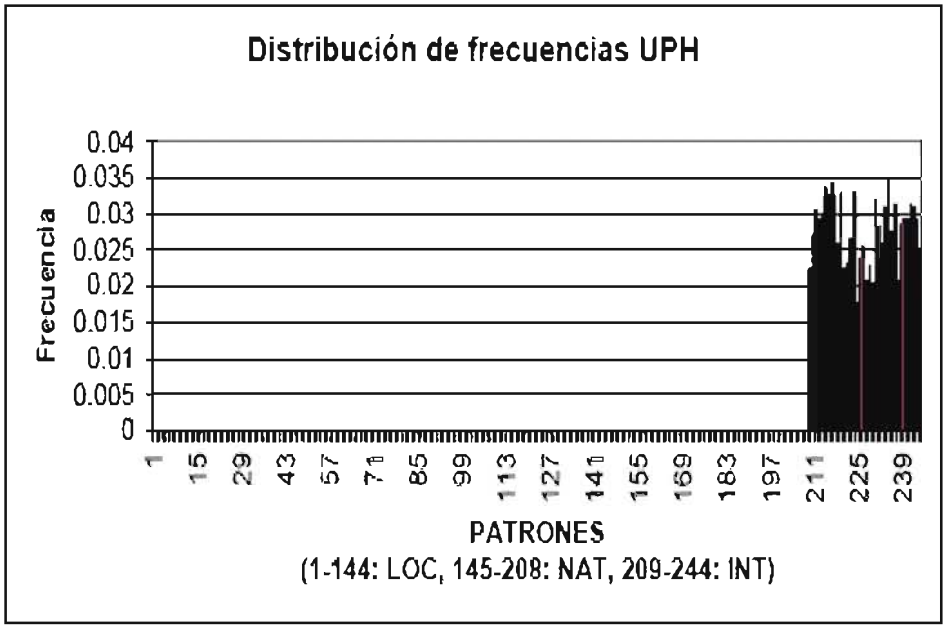


Gráfico 3.7: Distribución de frecuencias UPH experiencia 2

El gráfico 3.7 muestra el UPH del mismo usuario, se puede observar en el mismo que la distribución de frecuencias indica una tendencia a realizar llamadas internacionales solamente (patrones 209 a 244). En consecuencia la diferencia entre ambas distribuciones de frecuencias definida por la distancia Hellinger (H) es igual a: 0,82815. Analizando el detalle de llamadas de este usuario, se observa que el usuario solamente realizaba llamadas internacionales, pero un momento dado comenzó a realizar llamadas locales. Cuando la cantidad de llamadas locales modificó el CUP de la manera que se muestra en el gráfico, no es indicadora de fraude si el usuario paga su factura de llamadas internacionales; pero sí es indicadora de un sensible cambio de comportamiento en su patrón de consumo. La mayoría de las mismas siguen el patrón del caso que muestran los gráficos 3.6 y 3.7 en el cual debe haber varias llamadas fuera del patrón de comportamiento para que el sistema encuentre al usuario sospechoso. Esto es mucho más satisfactorio que lo obtenido en la experiencia 1 en la cual la alta sensibilidad mostraba usuarios como sospechosos simplemente por el hecho de haber realizado *una sola llamada* diferente.

CAPITULO IV
CONCLUSIONES Y RECOMENDACIONES

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

Este trabajo propone la construcción de un modelo de detección de fraude basada en la hipótesis que un cambio de comportamiento es susceptible de fraude, utilizando redes neuronales artificiales no supervisadas para la construcción de perfiles de usuario, en el marco de un análisis diferencial con enfoque de aprendizaje.

La solución propuesta, no solo ha demostrado ser viable y posible sino que además tiene aplicaciones adicionales no planteadas a priori, tales como la detección de cambios de comportamiento en los usuarios hacia modalidades que pueden hacer replantear los planes de tarifa definidos en la empresa u ofrecerle algún otro tipo de servicio al cliente.

La estructura que se definió con patrones LOC, patrones NAC y patrones INT mostró ser efectiva para representar el consumo de los usuarios y gráficamente se pudo observar y tener una idea de cuál era la forma en que un usuario se comportaba, tanto en su consumo reciente como en su consumo histórico. La idea de tener una distribución de frecuencia de los tipos de llamada que realizaba el usuario simplificó aún más el análisis y la posterior detección.

Experimentalmente se definió la estructura con 144 patrones LOC, 64 NAC y 32 INT que mostró ser suficiente para describir el comportamiento de los usuarios.
Las redes neuronales SOM han probado ser excelentes clasificadoras de las llamadas. Si se observan los gráficos presentados en el capitulo iii y el análisis correspondiente de cada uno de ellos se podrá ver que el espacio de llamadas se logró discretizar en grupos completamente representativos del consumo de los usuarios, teniendo en cuenta que los clientes de la empresa analizada son en su mayoría empresas y no usuarios individuales (el mayor uso se encuentra en la franja horaria laboral de 8 a 20hs. con duraciones de entre 0 y 7 minutos). Por lo

tanto, una vez definidos estos patrones, cada llamada que llegaba al sistema era muy fácilmente clasificada en el patrón que más se le parecía y "adaptada" al perfil a través de la ecuación de Grabec para predecir campos estocásticos en redes SOM; esto agregaba información a los perfiles, cambiando la distribución de frecuencia según correspondiese.

La información que se utilizó de las llamadas fueron el tipo de llamada, la hora y la duración. La primera dimensión determinó qué red neuronal se utilizaba para clasificarla y las dos siguientes definieron la entrada a la red neuronal correspondiente; se puede decir que la información presentada (normalizada a valores entre 0 y 1) fue suficiente para construir los perfiles y definir los patrones correspondientes.

4.2. RECOMENDACION

Se debe considerar nuevas investigaciones para el diseño, construcción e implementación de nuevos modelos para la detección de fraude en telefonías móviles utilizando RNA's y estudiar la posibilidad de su integración con el modelo neuronal del presente trabajo.

Para el presente trabajo se utilizo las redes SOM se recomienda estudiar la posibilidad de aplicación de otros métodos adicionales de aprendizaje para conseguir mejores resultados.

El modelo neuronal no pretende sustituir métodos o técnicas para la detección de fraude en telefonías móviles, sino más bien complementarlos.

REFERENCIA BIBLIOGRAFICA

Hilera J. R., Martínez V. 2000, Redes Neuronales Artificiales: Fundamentos, modelos y Aplicaciones, RA-MA Editorial, Madrid.

Gosset P., Hyland M., 1999. Classification, Detection and Prosecution of Fraud on Mobile.
Networks. http://www.esat.kuleuven.ac.be/cosic/aspect/papers/mobsummit.doc

Anderson, D.1995, Next-generation Intrusion Detection A Summary. SRI International Technical Report.

Northcutt, S. & Novak, J.2001, Detección de Intrusos. Guía Avanzada.
Segunda edición, editorial Pearson Educación S.A.

Catalina Gallego Alfredo, 2003, Introducción a las Redes Neuronales Artificiales, [Callin, 1996].

Fernández Javier & Sanguino, Peña (2002): Sistema de detección de Intrusos: Carencias actitudes tecnológicas, División de Seguridad de Germinus Solution.

Beveridge M, 1996. Self Organizing Maps.
http://www.dcs.napier.ac.uk/hci/martin/msc/node6.html

ASPeCT, 1996. *Definition of Fraud Detection Concepts, Deliverable D06*. 47 páginas.

Farley T., 2001. *Digital Wireless Basics*. Sitio con información sobre telefonía celular.
http://www.privateline.com/PCS/PCS.htm

Seymour B., 2000, *How Neural NetworkTechnology Can Tackle the Growing Telecom Fraud Problem*.
http://www.chipublishing.com/portal/backissues/pdfs/ISB_2000/ISB0503/ISB0503BS.pdf

Taniguchi M., Haft M., Hollmén J., Volker Tresp, 1998. *Fraud Detection In Communications Networks Using Neural And Probabilistic Methods*.
http://citeseer.nj.nec.com/taniguchi98fraud.html

Vesanto J., Alhoniemi E., 2000, *Clustering of the Self-Organizing Map*. IEEE transactions on neural networks, Vol 11, No. 3.

Burge P., Shawe-Taylor J., 1997. *Fraud Detection and Management in Mobile Telecommunications networks*, Department of Computer Science Royal Holloway, University of London. Vodafone, England. Siemens A. G. Proceedings of the 2nd European Conference on Security and Detection, IEE Conference publication 437, pp. 91-96, London

ANEXOS

ANEXOS
ANEXO A

CARACTERISTICAS DEL USO DEL SOFTWARE

El software Neuronal Trajan Network Simulator 6.0, cuyo entorno se observa en la Figura A.1, utilizado para implementacion del modelo neuronal permite el ingreso de los datos para el entrenamiento de la RNA en formato txt separado por tabulaciones, como se muestra en la figura A.2.

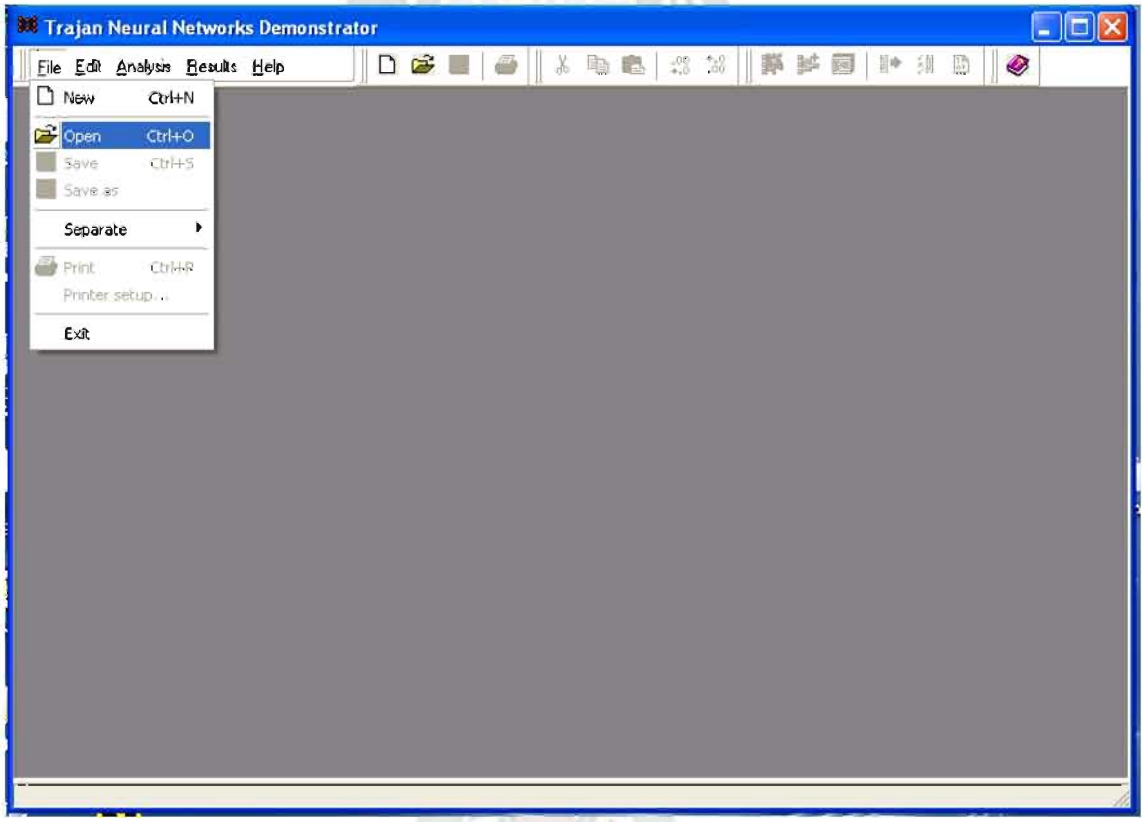


Figura A.1 Interfase del Trajan Neural Network Simulator 6.0.

El software permite la introducción manual de los datos de entrenamiento eligiendo File y luego new.

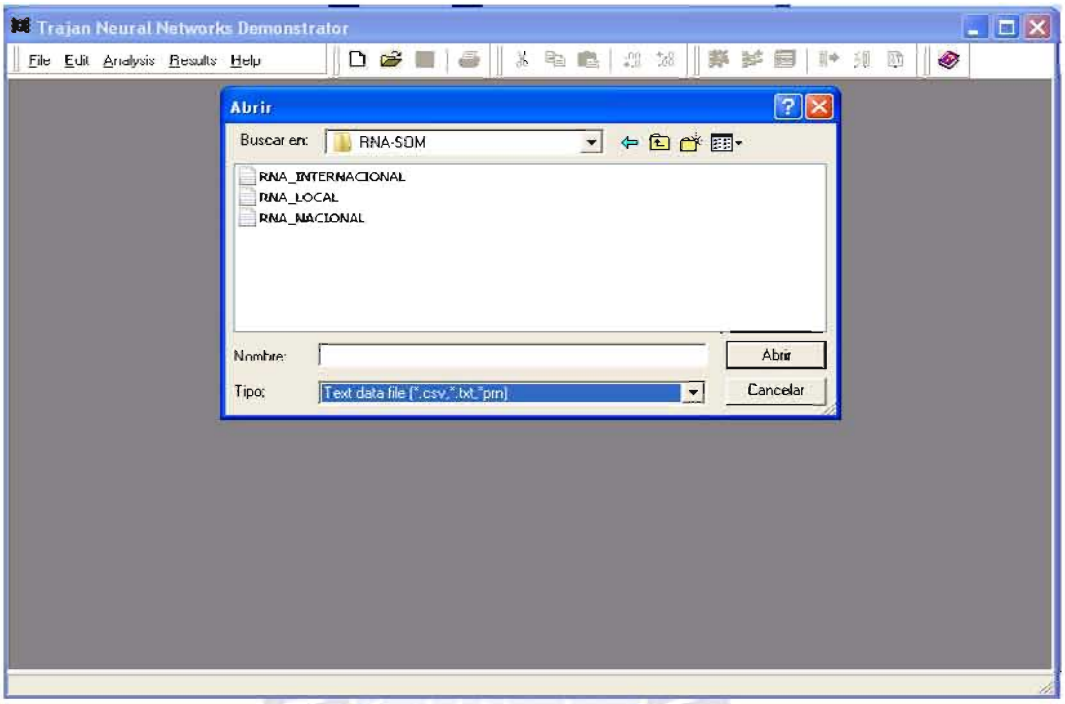


Figura A.2 Selección de los datos de entrenamiento para la RNA.

Luego se selecciona el botón Custon Network Designer para la creación de una RNA. Presionando luego el botón Variables para seleccionar las entradas (Independient) y la salida (Dependent) de la RNA. Ver Figura A.3.

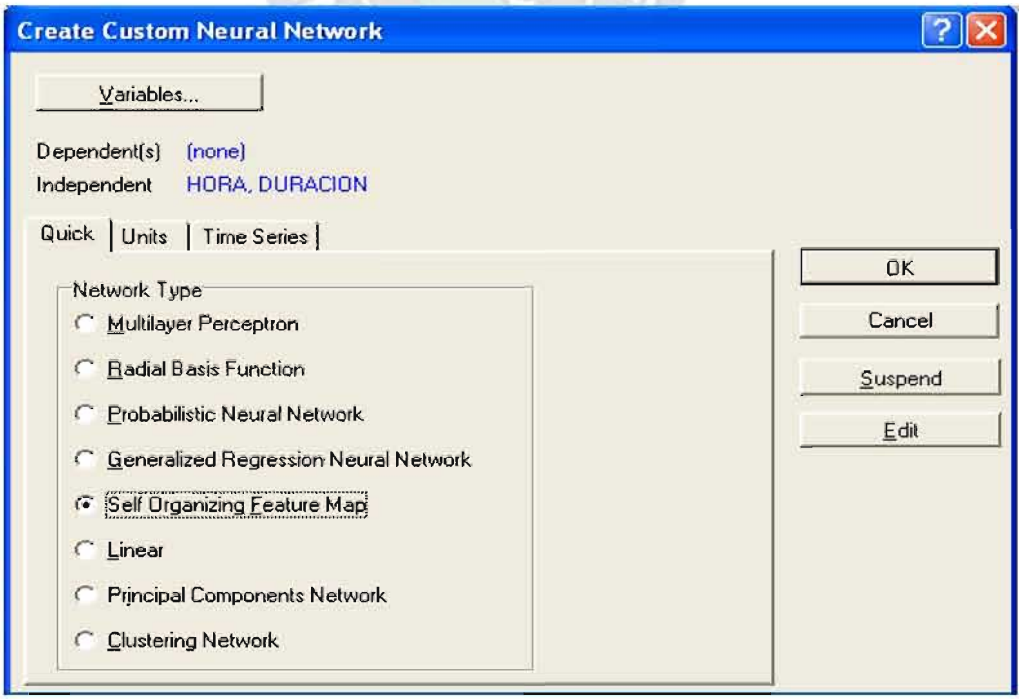


Figura A.3 Creación de RNA, definición de la Arquitectura de RNA.

Posteriormente se definen las iteraciones y tasa de aprendizaje, como se muestra en la figura A.4.

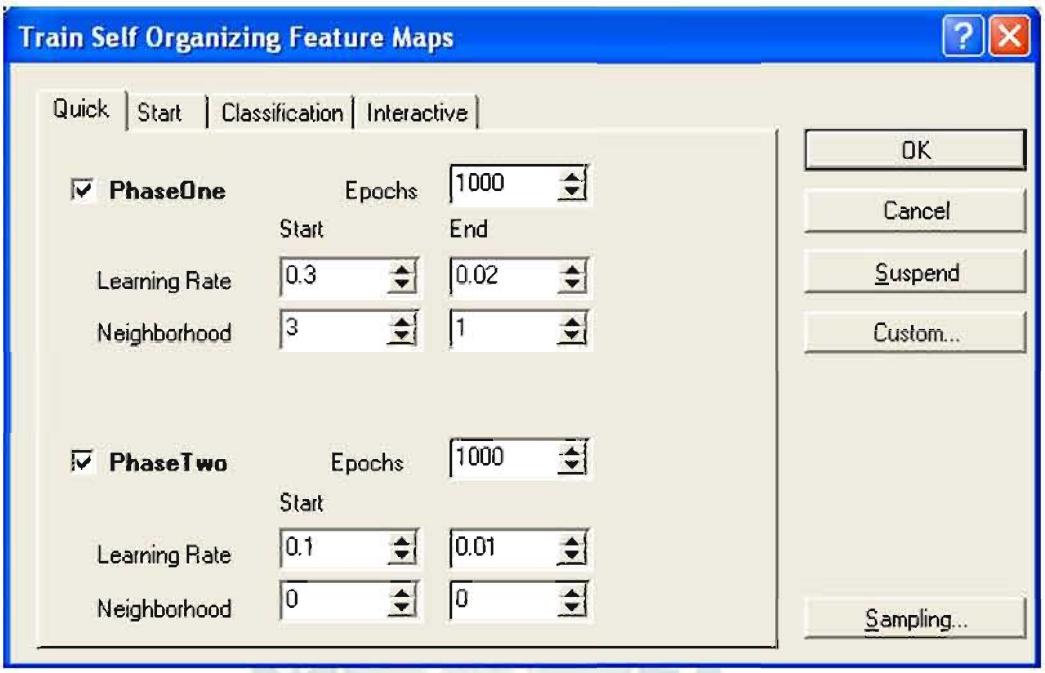


Figura A.4. Definición de iteraciones y tasa de aprendizaje.

Para finalmente presionar el botón OK, iniciando de esta forma el entrenamiento de la RNA, cuya duración depende de la capacidad de cómputo de la computadora utilizada y el numero de iteraciones que se eligieron. Ver Figura A.5

Figura A.5 Ventana de progreso del entrenamiento de la RNA.

En la figura A.6. Se puede observar el Mapa Topológico que genero el Software

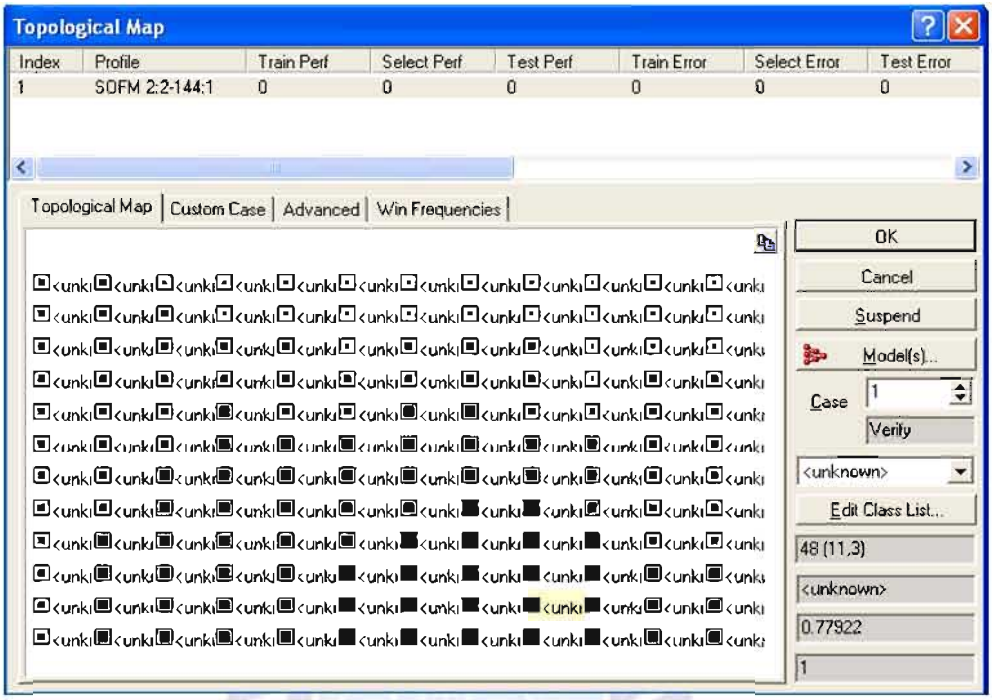


Figura A.6. Mapa Topológico

Posteriormente se puede observar el Modelo de Red Neuronal que se generó de acuerdo a los datos de entrada.

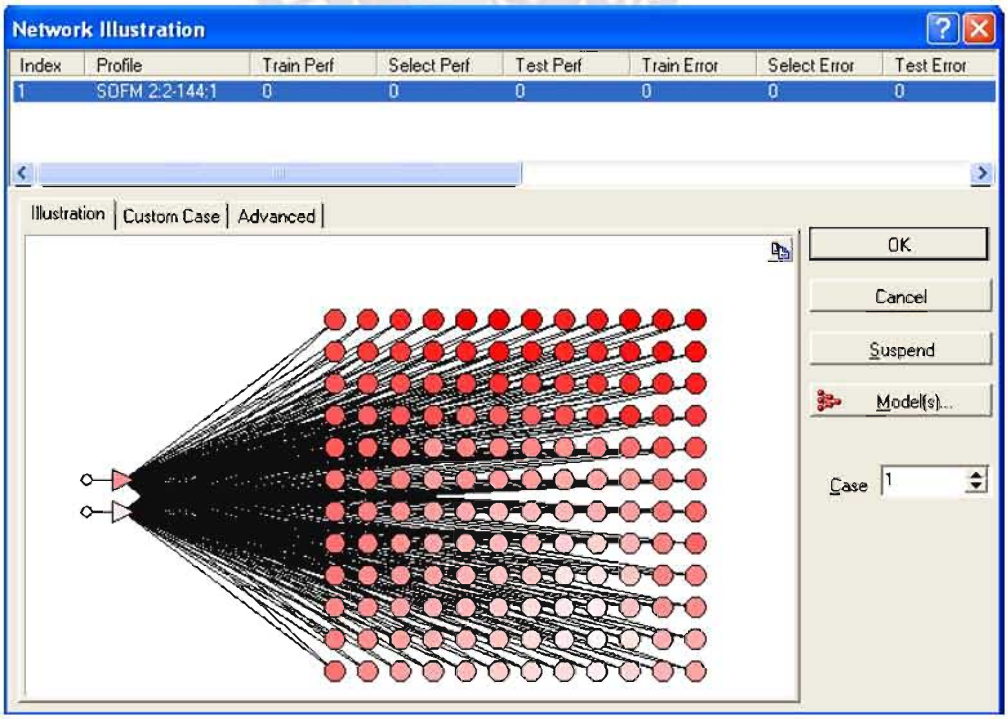


Figura A.7. Modelo de RNA

ANEXO B

Patrones definidos por cada una de las redes neuronales luego del Entrenamiento

PATRONES LOCALES

N°	HORA	DURACION
1	0.416667	0.120015
2	0.043741	0.951558
3	0.17658	0.83581
4	0.156875	0.325221
5	0.624875	0.277086
6	0.791667	0.044547
7	0.828942	0.554245
8	0.708333	0.064634
9	0.025681	0.368471
10	0.006242	1.000.000
11	0.583333	0.061421
12	0.5	0.204568
13	0.625907	0.525027
14	0.560102	0.333025
15	0.881239	0.494018
16	0.375	0.137896
17	0.833333	0.037046
18	0.786592	0.379005
19	0.368337	0.91393
20	0.625	0.10824
21	0.083278	0.834031
22	0.75	0.057901
23	0.088974	0.053845
24	0.168737	0.893429
25	0.625	0.218078
26	0.541667	0.12
27	0.110278	0.160563
28	0.41442	0.291351
29	0.405644	0.34312
30	0.275443	0.905062
31	0.458333	0.056739
32	0.174673	0.654854
33	0.545995	0.806322
34	0.666667	0.166686
35	0.708077	0.324604
36	0.224917	0.987805
37	0.092799	0.601243
38	0.8766	0.153459
39	0.756573	0.328048
40	0.41762	0.513437
41	0.505983	0.266872

42	0.541667	0.20395
43	0.75	0.133626
44	0.291667	0.042614
45	0.458333	0.108205
46	0.5	0.133585
47	0.541667	0.166667
48	0.375	0.033941
49	0.625	0.038019
50	0.105599	0.524099
51	0	0.033543
52	0.583333	0.120085
53	0.416667	0.057827
54	0.675057	0.404697
55	0.583333	0.174991
56	0.880232	0.637923
57	0.278272	0.467169
58	0.333333	0.100038
59	0.9406	0.232875
60	0.416644	0.172801
61	0.004041	0.324466
62	0.678649	0.265257
63	0.666667	0.103794
64	0.958333	0.041335
65	0.489078	0.439223
66	0.791667	0.121295
67	0.724237	0.592535
68	0.434559	0.949888
69	0.702329	0.84416
70	0.333333	0.064582
71	0.833333	0.219434
72	0.237411	0.188549
73	0.666667	0.033416
74	0.171331	0.839106
75	0.557677	0.495451
76	0.150152	0.092585
77	0.774694	0.701391
78	0.08887	0.92172
79	0.715	0.233197
80	0.628274	0.366674
81	0.918728	0.999995
82	0.95385	0.713698
83	0.916667	0.057045
84	0.708333	0.167487
85	0.876583	0.391028
86	0.094329	0.999998
87	0.59332	0.958387
88	0.917689	0.193157
89	0.205267	0.887631
90	0.041667	0.054619

91	0.423333	0.23262
92	0.50642	0.391012
93	0.000402	0.227413
94	0.874159	0.741281
95	0.000045	0.916041
96	0.5	0.1
97	0.497322	0.313841
98	0.032506	0.820723
99	0.173855	0.741816
100	0.391544	0.458172
101	0.3034	0.265315
102	0.958332	0.100296
103	0.000433	0.745396
104	0.458173	0.166667
105	0.777587	0.440966
106	0.868862	0.998566
107	0.097018	0.850826
108	0.915668	0.100016
109	0.557995	0.655797
110	0.408959	0.643947
111	0.5	0.03362
112	0.719576	0.997353
113	0.868378	0.269659
114	0.708333	0.109874
115	0.75	0.1
116	0.286542	0.613741
117	0.895848	0.562655
118	0.379593	0.776144
119	0.541667	0.054151
120	0.202731	0.033415
121	0.833333	0.100051
122	0.921313	0.275879
123	0.957913	0.38771
124	0.000004	0.614056
125	0.290041	0.12
126	0.337332	0.178154
127	0.029378	0.682531
128	0.081256	0.766203
129	0.930776	0.453382
130	0.875	0.033333
131	0.010263	0.1
132	0.000102	0.133526
133	0.837031	0.817946
134	0.25	0.041845
135	0.169996	0.573583
136	0.502076	0.58426
137	0.791667	0.19123
138	0.650696	0.592064
139	0.79077	0.242051

140	0.111537	0.907321
141	0.191931	0.812311
142	0.001601	0.490162
143	0.936735	0.859399
144	0.375	0.1

PATRONES NACIONALES

N°	HORA	DURACION
1	0.376018	0.168956
2	0.673286	0.410396
3	0.916667	0.041369
4	0.942427	0.296879
5	0.833336	0.120091
6	0.583589	0.064534
7	0.749997	0.128512
8	0.458333	0.037045
9	0.643331	0.996948
10	0.791667	0.120306
11	0.00869	0.581652
12	0.372285	0.232881
13	0.455668	0.17946
14	0.915559	0.646052
15	0.875	0.046408
16	0.955779	0.129569
17	0.929179	0.233378
18	0.958332	0.191208
19	0.958333	0.033464
20	0.416666	0.033333
21	0.91666	0.175211
22	0.873749	0.476504
23	0.416667	0.098509
24	0.79171	0.244627
25	0.873265	1
26	0.291656	0.041541
27	0.603198	0.525312
28	0.666667	0.10149
29	0.846024	0.295222
30	0.841306	0.717394
31	0.374989	0.108083
32	0.090201	1
33	0.948992	0.432418
34	0.614844	0.348353
35	0.003268	0.033907
36	0.833333	0.061453
37	0.603444	0.907656
38	0.918011	0.486237
39	0.949965	0.88854
40	0.627411	0.14722

41	0.874959	0.22334
42	0.791667	0.037814
43	0.541667	0.034906
44	0.664957	0.033878
45	0.176409	0.046571
46	0.856534	0.333836
47	0.517907	0.262286
48	0.54295	0.698061
49	0.051248	0.84462
50	0.005312	0.148722
51	0.823333	0.177957
52	0.414976	0.464157
53	0.702759	0.255123
54	0.875	0.149844
55	0.582934	0.104064
56	0.744901	0.615677
57	0.625	0.090873
58	0.708333	0.056637
59	0.058246	0.319713
60	0.5	0.063133
61	0.333333	0.033928
62	0.930177	0.787899
63	0.61658	0.205661
64	0.937545	0.537037

PATRONES INTERNACIONALES

HORA	DURACION
0.010052	0,104071
0.957958	0,033551
0.371844	0,11971
0.532821	0,509725
0.893357	0,164537
0.569367	0,23681
0.630093	0,316
0.662728	0,172234
0.722712	0,033589
0.172583	0,043936
0.482099	0,152262
0.009922	0,355681
0.188025	0,645894
0.939822	0,466382
0.566731	0,034614
0.096951	0,975024
0.871037	0,740844
0.820292	0,649491
0.009961	0,600516
0.025044	0,061434
0.841131	0,998922

0.491031	0,034613
0.911812	0,35198
0.502443	0,69361
0.658614	0,425429
0.617711	0,154931
0.73817	0,591849
0.797576	0,033333
0.472464	0,998538
0.586649	0,556765
0.803976	0,435338
0.884635	0,063903
0.823191	0,183571
0.474916	0,376312
0.318363	0,267793
0.10439	0,12889

ANEXO C

PANTALLAS DEL PROTOTIPO