26-11-2024

Adrián Jiménez Benítez

Universidad de almeria

Práctica 3

Redes Bayesianas usando la herramienta Hugin y Python

Contenido

[Ilustraciones 2](#_Toc183605334)

[Enunciado 3](#_Toc183605335)

[Ejercicio 1 3](#_Toc183605336)

[Apartado A: 3](#_Toc183605337)

[Respuesta a preguntas: 5](#_Toc183605338)

[Apartado B: 6](#_Toc183605339)

[Apartado C: 7](#_Toc183605340)

[Apartado D: 7](#_Toc183605341)

# Ilustraciones

[Ilustración 1: Red Bayesiana 4](#_Toc183605322)

[Ilustración 2: Tabla Request 4](#_Toc183605323)

[Ilustración 3: Tabla Firewall 4](#_Toc183605324)

[Ilustración 4: Tabla Attacks 4](#_Toc183605325)

[Ilustración 5: Tabla Recovery 4](#_Toc183605326)

[Ilustración 6: Tabla System 5](#_Toc183605327)

[Ilustración 7: Probabilidades 5](#_Toc183605328)

[Ilustración 8: Probabilidad cortafuegos desactivado 6](#_Toc183605329)

[Ilustración 9: Creación de la red Bayesiana 6](#_Toc183605330)

[Ilustración 10: Código Python 7](#_Toc183605331)

[Ilustración 11: Nombres de Variables 7](#_Toc183605332)

[Ilustración 12: Variables Marginalmente independientes 8](#_Toc183605333)

# Enunciado

*Un servidor web recibe cada hora una media de 1000 peticiones HTTP,  
de las cuales se estima que 100 son ataques. De éstos, se estima que 50  
pueden suponer un fallo grave para la integridad del sistema. Se instala  
un cortafuegos que, según especificaciones del fabricante, no conseguirá  
bloquear 1 de cada 4 ataques. Además, dicho cortafuegos se actualizará  
una vez cada hora, lo cual tardará 3 minutos y durante este tiempo no  
estará activo. Por otro lado, se instalará un sistema de recuperación que  
estará en funcionamiento el 75 % del tiempo y que evitará la mitad de los  
fallos graves.*

# Ejercicio 1

## Apartado A:

*Utilizando la herramienta Hugin, modela este escenario como una  
red Bayesiana. ¿Cuál es la probabilidad de que, en cualquier momen-  
to, se produzca un fallo grave? ¿Y si se desconecta completamente el  
cortafuegos?*

Para comenzar se va a ver los nodos y la probabilidad que tiene cada nodo.

* Nodo Request: Probabilidad de que la petición sea un ataque. Como muestra el enunciado la probabilidad de una petición sea un ataque es del 10%.
* Nodo Firewall: Probabilidad de que el firewall esté en funcionamiento. Está en funcionamiento el 95% del tiempo, porque el cortafuegos se actualizará una vez cada hora durante 3 minutos . Por tanto, el funcionamiento del firewall será .
* Nodo Attack: Probabilidad de que el ataque cruce el cruce el firewall o no.
* Nodo Recovery: Probabilidad de que el sistema de recuperación esté activo o no (75%).
* Nodo System: Probabilidad de que el sistema tenga un fallo grave o no.

#### Red Bayesiana

En este apartado, se mostrará la red bayesiana implementada en Hugin:

Diagrama

Descripción generada automáticamente

Ilustración 1: Red Bayesiana

#### Tabla de Probabilidades

A continuación, se verá como son las tablas generadas por Hugin al ponerle las probabilidades de cada caso.

##### Tabla request:



Ilustración 2: Tabla Request

##### Tabla Firewall:

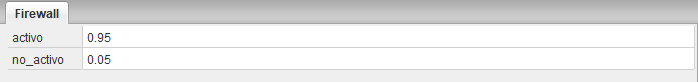


Ilustración 3: Tabla Firewall

##### Tabla Attacks:

Tabla

Descripción generada automáticamente con confianza media

Ilustración 4: Tabla Attacks

##### Tabla Recovery:



Ilustración 5: Tabla Recovery

##### Tabla System:

Tabla

Descripción generada automáticamente

Ilustración 6: Tabla System

### Respuesta a preguntas:

Para responder a las preguntas, se pulsa el rayo que se encuentra en la parte de arriba del programa, y sale la siguiente pantalla:

Diagrama

Descripción generada automáticamente

Ilustración 7: Probabilidades

#### *¿Cuál es la probabilidad de que, EN CUALQUIER MOMENTO, ¿SE PRODUZCA UN FALLO GRAVE?*

La probabilidad de que se produzca un fallo grave en cualquier momento es del 0.9%, tal y como muestra la imagen.

#### *¿Y si se desconecta completamente el cortafuegos?*

Diagrama

Descripción generada automáticamente

Ilustración 8: Probabilidad cortafuegos desactivado

Cuando no está el firewall activado, la probabilidad de que se produzca un fallo grave es del 3.13%.

## Apartado B:

*Utilizando el paquete PyAgrum, especifica el código en Python que permita definir la misma red y calcular las probabilidades del apartado anterior.*

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 9: Creación de la red Bayesiana

Para obtener este resultado, primero se creó la misma red Bayesiana, como se puede observar en la imagen anterior. Posteriormente, el código desarrollado permite calcular las probabilidades que previamente ha generado la aplicación Hugin, tal como se muestra en la siguiente imagen.

*Interfaz de usuario gráfica, Texto

Descripción generada automáticamente*

Ilustración 10: Código Python

## Apartado C:

*Define una función que tome como parámetro de entrada una red Bayesiana y devuelva una lista con los nombres de las variables.*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 11: Nombres de Variables

La función *“get\_variable\_names(bn)”* está diseñada para obtener los nombres de las variables que componen una red bayesiana, representada aquí por el objeto bn. Este objeto contiene tanto los nodos de la red como las variables asociadas.

## Apartado D:

*Escribe el código que indique, para cada par de variables, si son marginalmente independientes o no.*

Texto

Descripción generada automáticamente

Ilustración 12: Variables Marginalmente independientes

La función *“check\_marginal\_independence(bn)”* evalúa la independencia marginal entre todos los pares de variables en una red bayesiana, representada por el objeto bn.