ECG1

DM de la semaine du 14/10.

Exercice 1 (Modélisation probabiliste d'un système de détection d'intrusion avancé) Une entreprise de cybersécurité a développé un système de détection d'intrusion (IDS) multicouche pour protéger les réseaux de ses clients. Le système fonctionne comme suit :

- Couche 1 (C1) : Analyse du trafic réseau, détecte 75% des intrusions.
- Couche 2 (C2) : Analyse comportementale, détecte 60% des intrusions non repérées par C1.
- Couche 3 (C3) : Intelligence artificielle, détecte 80% des intrusions ayant échappé à C1 et C2.

De plus :

- Le système peut générer des fausses alertes avec une probabilité de 5% pour C1, 3% pour C2, et 2% pour C3.
- On estime qu'il y a en moyenne 1000 tentatives d'accès par jour, dont 1% sont des intrusions réelles.
- Chaque fausse alerte coûte 200 à l'entreprise en temps et en ressources.
- Chaque intrusion non détectée coûte en moyenne 100 000.

1. Modélisez mathématiquement le système IDS. Définissez clairement les événements et les probabilités associées.
2. Calculez la probabilité qu'une intrusion réelle ne soit pas détectée par le système.
3. Calculer la probabilité qu'une alerte soit déclenchée pour un accès quelconque.
4. On observe qu'une alerte a été déclenchée. Quelle est la probabilité qu'il s'agisse d'une vraie intrusion ?
5. Soit A l'événement "une alerte est déclenchée par C1" et B l'événement "une alerte est déclenchée par C2". Déterminez si A et B sont indépendants. Justifiez votre réponse.
6. L'entreprise envisage d'ajouter une quatrième couche (C4) qui détecterait 90% des intrusions ayant échappé aux trois premières couches, mais qui augmenterait le taux de fausses alertes de 1% et coûterait 5000 par jour à maintenir. Déterminez si cet investissement est rentable à long terme.

1. Modélisation : Soit I l'événement "il y a une intrusion" $P(I) = 0,01$

   Soit D_i l'événement "détection par la couche i" $P_{D_1}(I) = 0,75$ $P_{D_2}(I \cap \overline{D_1}) = 0,60$ $P_{D_3}(I \cap \overline{D_1} \cap \overline{D_2}) = 0,80$

   Soit F_i l'événement "fausse alerte de la couche i" $P_{F_1}(\overline{I}) = 0,05$ $P_{F_2}(\overline{I}) = 0,03$ $P_{F_3}(\overline{I}) = 0,02$

2. Probabilité qu'une intrusion ne soit pas détectée : $P_{\overline{D_1} \cap \overline{D_2} \cap \overline{D_3}}(I) = (1 - 0,75) \times (1 - 0,60) \times (1 - 0,80) = 0,02$

3. Probabilité qu'une alerte soit déclenchée (formule des probabilités totales) : $P(Alerte) = P_{Alerte}(I)P(I) + P_{Alerte}(\overline{I})P(\overline{I})$

   $P_{Alerte}(I) = 1 - P_{\overline{D_1} \cap \overline{D_2} \cap \overline{D_3}}(I) = 0,98$

   $P_{Alerte}(\overline{I}) = 1 - (1 - 0,05)(1 - 0,03)(1 - 0,02) = 0,0975$

   $P(Alerte) = 0,98 \times 0,01 + 0,0975 \times 0,99 = 0,1065$

4. Probabilité qu'une alerte corresponde à une vraie intrusion : $P_I(Alerte) = \frac{P_{Alerte}(I)P(I)}{P(Alerte)} = \frac{0,98 \times 0,01}{0,1065} \approx 0,0920$

5. Indépendance de A et B : $P(A) = P(I)P_{D_1}(I) + P(\overline{I})P_{F_1}(\overline{I}) = 0,01 \times 0,75 + 0,99 \times 0,05 = 0,0570$
   $P(B) = P(I)P_{\overline{D_1}}(I)P_{D_2}(I \cap \overline{D_1}) + P(\overline{I})P_{F_2}(\overline{I}) = 0,01 \times 0,25 \times 0,60 + 0,99 \times 0,03 = 0,0312$

   $P(A \cap B) = P(I)P_{D_1}(I)P_{F_2}(\overline{I}) + P(\overline{I})P_{F_1}(\overline{I})P_{F_2}(\overline{I}) = 0,01 \times 0,75 \times 0,03 + 0,99 \times 0,05 \times 0,03 = 0,001714$

   $P(A) \times P(B) = 0,0570 \times 0,0312 = 0,001778$

   Comme $P(A \cap B) \neq P(A) \times P(B)$, A et B ne sont pas indépendants.

6. Coût moyen quotidien : Coût des intrusions non détectées : $1000 \times 0,01 \times 0,02 \times 100000 = 20000$ Nombre moyen de fausses alertes : $1000 \times 0,99 \times 0,0975 = 96,525$ Coût des fausses alertes : $96,525 \times 200 = 19305$ Coût total moyen : $20000 + 19305 = 39305$ par jour

7. Rentabilité de la quatrième couche : Nouvelle probabilité de non-détection : $0,02 \times 0,10 = 0,002$ Nouveau coût des intrusions non détectées : $1000 \times 0,01 \times 0,002 \times 100000 = 2000$ Économie réalisée sur les intrusions : $20000 - 2000 = 18000$

   Nouveau taux de fausses alertes : $0,0975 + 0,01 = 0,1075$ Nouveau coût des fausses alertes : $1000 \times 0,99 \times 0,1075 \times 200 = 21285$ Augmentation du coût des fausses alertes : $21285 - 19305 = 1980$

   Bilan : $18000 - 1980 - 5000 = 11020$ d'économie par jour

   L'investissement est donc rentable à long terme.

Exercice 2 Soient $A_1, \ldots, A_n$ des évènements mutuellement indépendants. Montrer que la probabilité qu'aucun des A_i ne soit réalisé est inférieure à

$$\exp\left(-\sum_{i=1}^{n} \mathrm{P}\left(A_i\right)\right)$$

Indication : on pourra montrer que, pour tout réel x, on a $1 - x \leq e^{-x}$

Commençons par montrer l'inégalité indiquée dans l'énoncé. Soit $x \in \mathbb{R}$. Considérons la fonction

$$f : x \mapsto e^{-x} - (1 - x)$$

On a $f(0) = 0$ et f est dérivable sur $\mathbb{R}$ de dérivée $f' : x \mapsto -e^{-x} + 1$. Dressons le tableau de signe de f :

x	$-\infty$		0		$+\infty$
$f'(x)$		$-$	0	$+$	
f	$-\infty$	$\searrow$	0	$\nearrow$	$-\infty$

Par conséquent, on a bien $1 - x \leq e^{-x}$ pour tout réel x.
On a, par indépendance mutuelle,

$$\mathrm{P}\left(\bigcap_{i=1}^{n} \overline{A_i}\right) = \prod_{i=1}^{n} \mathrm{P}\left(\overline{A_i}\right) = \prod_{i=1}^{n} (1 - \mathrm{P}(A_i))$$

En utilisant l'inégalité donnée dans l'indication, on obtient :

$$\prod_{i=1}^{n} (1 - \mathrm{P}(A_i)) \leq \prod_{i=1}^{n} e^{-\mathrm{P}(A_i)} = e^{-\sum_{i=1}^{n} \mathrm{P}(A_i)}$$

Ainsi, on conclut que :

$$\boxed{\mathrm{P}\left(\bigcap_{i=1}^{n} \overline{A_i}\right) \leq e^{-\sum_{i=1}^{n} \mathrm{P}(A_i)}}$$

Exercice 3 (Approximation du nombre d'or)

On appelle nombre d'or et on note ϕ la racine réelle positive du polynôme $X^2 - X - 1$.

1. Exprimez ϕ et justifier, sans calculatrice, que $1 < \phi < 2$.
2. On considère la suite (u_n) définie sur $\mathbb{N}^*$ par

$$u_1 = \sqrt{1},\ u_2 = \sqrt{1 + \sqrt{1}},\ u_3 = \sqrt{1 + \sqrt{1 + \sqrt{1}}}$$

et ainsi de suite,

$$u_n = \sqrt{1 + \cdots + \sqrt{1+\sqrt{1}}}$$

avec n radicaux.
Exprimer, pour tout entier n supérieur ou égal à 1, u_{n+1} en fonction de u_n et donner f tel que $u_{n+1} = f(u_n)$

3. Ecrire un programme en Python intitulé `suite`, d'argument une variable `n` et qui calcule le n-ème terme de la suite.
4. Dresser le tableau de variation de f et résoudre l'équation $f(x) = x$.
5. Montrez que pour tout $n \geq 1$,
$$0 \leq u_n \leq \phi.$$
6. Montrer que, pour tout entier $n \geq 1$,
$$|u_{n+1} - \phi| \leq \frac{1}{2}|u_n - \phi|.$$
Indication : commencez par montrer, en le justifiant, que $u_{n+1} - \phi = \sqrt{1+u_n} - \sqrt{1+\phi}$ et utilisez l'expression conjuguée.
7. En déduire que, pour tout $n \geq 1$,
$$|u_n - \phi| \leq \frac{1}{2^{n-1}}$$
8. Conclure sur la convergence de la suite u_n.
9. On rappelle ce qui a été décrit dans l'exo 3 : on dit qu'un nombre x est une approximation d'un nombre α à 10^{-p} près si $|x - \alpha| \leq 10^{-p}$. Par exemple, $3,14$ est une approximation à 10^{-2} près de π.

   Soit $p \in \mathbb{N}^*$. A l'aide de la question 7, déterminez n tel que u_n est une approximation de ϕ à 10^{-p} près.
10. Application numerique : on pose $p = 1000$, et on suppose que l'ordinateur met 1 milliseconde pour calculer 100 termes de la suite $(u_n)_{n\in\mathbb{N}}$. Combien de temps l'ordinateur va-t-il mettre pour obtenir une précision à 1000 décimales du nombre d'or ?
    Comparez avec l'algorithme de l'exercice de la semaine passée.

1. On trouve que le polynôme a deux racines (avec $\Delta = 5$) qui sont $\frac{1-\sqrt{5}}{2}$ et $\frac{1+\sqrt{5}}{2}$. Or $\sqrt{5} > 1$ donc la première racine est négative et on a
$$\boxed{\phi = \frac{1+\sqrt{5}}{2}}$$
   On peut procéder par encadrement :
   Comme $2 < \sqrt{5} < 3$ (puisque $4 < 5 < 9$), on a $3 < 1 + \sqrt{5} < 4$ et donc $\frac{3}{2} < \frac{1+\sqrt{5}}{2} < 2$.
   Comme $\frac{3}{2} > 1$, on a l'inégalité voulue.
   On peut aussi poser $P(x) = x^2 - x - 1$ et voir que $P(1) = -1$ tandis que $P(2) = 1$: la racine est donc entre 1 et 2, d'où
$$\boxed{1 < \phi < 2}$$
2. On observe qu'il s'agit à chaque fois d'ajouter 1 et de prendre la racine du tout. Ainsi, $u_{n+1} = \sqrt{1+u_n} = f(u_n)$ avec $f : x \mapsto \sqrt{1+x}$.
3. On peut proposer le programme suivant :

```
from math import sqrt
def suite(n):
   u=1
   for k in range(n):
     u=sqrt(1+u)
   return u
```

4. La fonction f est définie sur $[-1, +\infty[$, et dérivable sur $]-1, +\infty[$ avec $f'(x) = \frac{1}{2\sqrt{1+x}}$.

Donc f est croissante (strictement) d'où le tableau ci dessous (où j'ai déjà ajouté ϕ mais la preuve vient juste après...)

x	-1		ϕ		$+\infty$
$f'(x)$	$\|\|$		$+$		
$f(x)$	0	$\nearrow$	ϕ	$\nearrow$	$+\infty$

Résolvons maintenant l'équation $f(x) = x$, c'est à dire $\sqrt{1+x} = x$

Notons déjà que l'équation est définie sur $[-1, +\infty[$, et comme pour tout $x \in [-1, +\infty[$, $\sqrt{1+x} \geq 0$, on a forcément $x \geq 0$.

Ainsi, on résout l'équation sur $\mathbb{R}_+$, où tous les termes sont positifs et où on peut donc utiliser l'injectivité de $x \mapsto x^2$ sur $\mathbb{R}_+$:

$$\sqrt{1+x} = x \iff 1 + x = x^2 \iff x^2 - x - 1 = 0$$

On revient ainsi à la question 1, où l'unique racine positive de ce polynôme est ϕ.

En conclusion : $\boxed{f(x) = x \iff x = \phi}$

5. Par récurrence, montrons que $0 \leq u_n \leq \phi$.

   Initialisation : on a déjà $u_1 = 1 \in [0, \phi]$

   Hérédité : Soit $n \geq 1$, supposons que $0 \leq u_n \leq \phi$.

   Comme f est croissante, on a $f(0) \leq f(u_n) \leq f(\phi)$.

   Or $f(0) = \sqrt{1} = 1 \geq 0$, $f(u_n) = u_{n+1}$ et $f(\phi) = \phi$ (cf question précédente), donc

   $$0 \leq u_{n+1} \leq \phi$$

   la propriété est héréditaire.

   $\boxed{\text{Conclusion : } \forall n \geq 1, u_n \in [0, \phi]}$

6. On suit l'indication : $u_{n+1} = \sqrt{1+u_n}$ par définition, et on a vu à la question 4 que $\sqrt{1+\phi} = \phi$ (puisque $f(\phi) = \phi$), donc

$$\begin{aligned} u_{n+1} - \phi &= \sqrt{1+u_n} - \sqrt{1+\phi} \\ &= \frac{(\sqrt{1+u_n} - \sqrt{1+\phi})(\sqrt{1+u_n} + \sqrt{1+\phi})}{\sqrt{1+u_n} + \sqrt{1+\phi}} \\ &= \frac{1+u_n-1-\phi}{\sqrt{1+u_n} + \sqrt{1+\phi}} \end{aligned}$$

Ainsi $|u_{n+1} - \phi| = |\frac{u_n - \phi}{\sqrt{1+u_n}+\sqrt{1+\phi}}|$

Or $\sqrt{1+u_n} \geq 1$ et $\sqrt{1+\phi} \geq 1$, donc $\sqrt{1+u_n} + \sqrt{1+\phi} \geq 2$ et finalement

$$|u_{n+1} - \phi| \leq |\frac{u_n - \phi}{2}|$$

7. C'est une récurrence assez simple si on utilise bien la propriété précédente :

Initialisation : pour $n = 1$, on a, d'après la première question, $|u_1 - \phi| = |1 - \phi| \leq 1 = \frac{1}{2^{1-1}}$

Héréditié : Soit $n \geq 1$. Supposons que $|u_n - \phi| \leq \frac{1}{2^{n-1}}$, alors

$$|u_{n+1} - \phi| \leq \frac{1}{2}|u_n - \phi| \leq \frac{1}{2}\frac{1}{2^{n-1}} = \frac{1}{2^n}$$

La propriété est donc héréditaire et l'inégalité est vérifiée pour tout n.

8. Comme $\lim \frac{1}{2^n} = 0$ et que $0 \leq |u_n - \phi| \leq \frac{1}{2^n}$, on en déduit, par le théorème des gendarmes, que $\lim |u_n - \phi| = 0$, c'est à dire $\boxed{\lim u_n = \phi}$.

9. Il suffit de trouver n tel que $\frac{1}{2^n} \leq 10^{-p}$.

C'est à dire $2^n \geq 10^p$, d'où $\ln(2^n) \geq \ln(10^p)$, c'est à dire $n\ln(2) \geq p\ln(10)$.

Conclusion : avec $n \geq p\frac{\ln(10)}{\ln(2)}$, on a une approximation à 10^{-p} de ϕ.

10. pour $p = 1000$, on obtient $n \simeq 3322$. Il nous faudra à peine plus de 33 millisecondes pour avoir 1000 décimales exactes de notre nombre d'or.... Ce qui est définitivement plus rapide que ce qu'on avait dans l'exercice précédent !!!