Universitat de Girona
Escola Politècnica Superior

Projecte fi de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Tècniques d'Intel·ligència artificial per calcular la robustesa de xarxes

Document: Resum

Alumne: Martí Madrenys Masferrer

Tutor: Eusebi Calle Ortega
Departament: ARQUITECTURA I TECNOLOGIA DE COMPUTADORS
Àrea: ARQUITECTURA I TECNOLOGIA DE COMPUTADORS

Convocatòria (mes/any): Juny 2022

1 Conceptes previs

L'NRS és un simulador de xarxes propietat del grup de recerca BCDS. Amb aquesta eina es poden visualitzar aquestes xarxes, calcular-ne mètriques i calcular la robustesa que presenten davant de certs atacs.

Un atac a una xarxa implica l'eliminació un o més nodes o arestes. Un atac genera sempre un subconjunt de la xarxa original, aquesta xarxa resultant tindrà igual o menys connectivitat.

En termes de xarxes, la robustesa es pot definir com la capacitat d'una xarxa de continuar funcionant correctament davant de fallades o atacs. La robustesa es calcula usant conceptes de la teoria de grafs, més concretament, de connectivitat de grafs. [Marzo 2019].

Una mètode de càlcul d'aquesta mètrica podria ser el proposat per Trajanovski et alt. [Trajanovski 2013]:

$$R = \sum_{n}^{k=1} s_k t_k$$

On s_k representa el pes de la mètrica i t_k el valor de la mètrica k.

El grup BCDS va fer un estudi per decidir quines mètriques tenir en compte i va dissenyar un mètode automatitzat pel càlcul de robustesa. [Marzo 2018].

L'algorisme que fa aquest càlcul és computacionalment costós, per obtenir resultats d'alguns *datasets* grans pot necessitar de l'ordre de fins a **varis dies**.

2 Objectius

En aquest Treball de Final de Grau es pretén assolir els següents objectius:

1. Estudiar l'estat de l'art del camp d'intel·ligència artificial, més concretament de l'aprenentatge supervisat.
2. Desenvolupar models d'intel·ligència artificial que permetin calcular la robustesa de xarxes.
3. Estudiar si generant models específics per cada atac millora la precisió dels càlculs.
4. Comparar empíricament el cost benefici d'usar aquests models respecte els algoritmes clàssics.
5. Crear una interfície per poder usar els models i fer prediccions de manera senzilla.

3 Estudi i decisions

Durant la fase d'estudi del projecte es revisen models existents que pugin solucionar el problema plantejat.

Els candidats inicials han sigut:

- *Linear Regression* (Regressió Linear general)
- *K-Nearest Neighbours* (k-veïns propers)
- *Decision Tree / Random forest* (arbres de decisió i boscos)
- *Neural Network* (Xarxa neuronal).

Un cop triats els candidats, es pren la decisió d'utilitzar una llibreria d'alt nivell que ja implementa models per tots els candidats. Aquesta llibreria és la SciKit Learn de Python [Pedregosa 2011].

També es decideix que s'utilitzarà l'entorn Jupyter Notebook pel fet que facilita molt treballar amb funcions pesades i executar petites funcions en models ja entrenats sobre la marxa.

4 Implementació

Pel que respecte als candidats triats, es fa una implementació en dues fases:

1. La primera fase és la comuna, en aquesta fase comuna es farà servir la eina d'importació de dades de l'NRS i es farà un primer tractament de dades.
2. En la segona fase, s'implementaran les característiques específiques de cada model candidat.

4.1 Primers resultats

Els candidats implementats inicialment s'avaluen amb el mateix mètode, a partir de l'error relatiu. Amb aquest sistema podem comparar els primers resultats entre els models que es poden trobar a la taula 1.

Candidat	Error Relatiu Mitja
Linear Regression	5.10%
KNN	2.83%
Decission Tree	2.80%
Random Forest	2.36%
Neural Network	4.62%

Taula 1: Taula amb els resultats dels candidats inicials implementats.

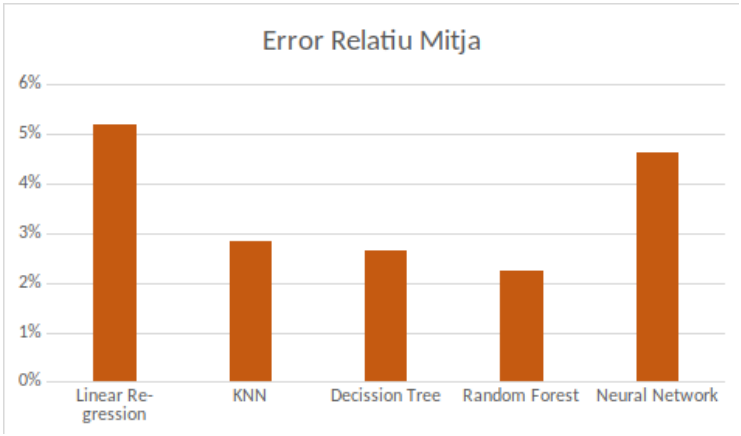

Figura 1: Gràfic de barres on es poden veure els diferents candidats a l'eix X i els error mitjà a l'eix Y.

En aquest punt i donats els bons resultats inicials es decideix no implementar més candidats i centrar-se en el millor model fins ara, el *Random Forest*. A aquest model se li aplicaran una serie de millores.

4.2 Millores

La primera millora que s'intenta aplicar és una optimització d'hiperparàmetres, aquests són unes característiques intrínseques de cada model que tot i tenir un valor per defecte es poden modificar per intentar obtenir millors resultats al fer prediccions.

El resultat del model optimitzat no millora l'original, per aquest motiu, es descarta aquesta millora.

La segona millora incorpora més mètriques al *dataset* i un el còmput d'un estimador de generalització del mode. Amb aquesta millora es pot obtenir l'ordre d'importància de les mètriques segons el model

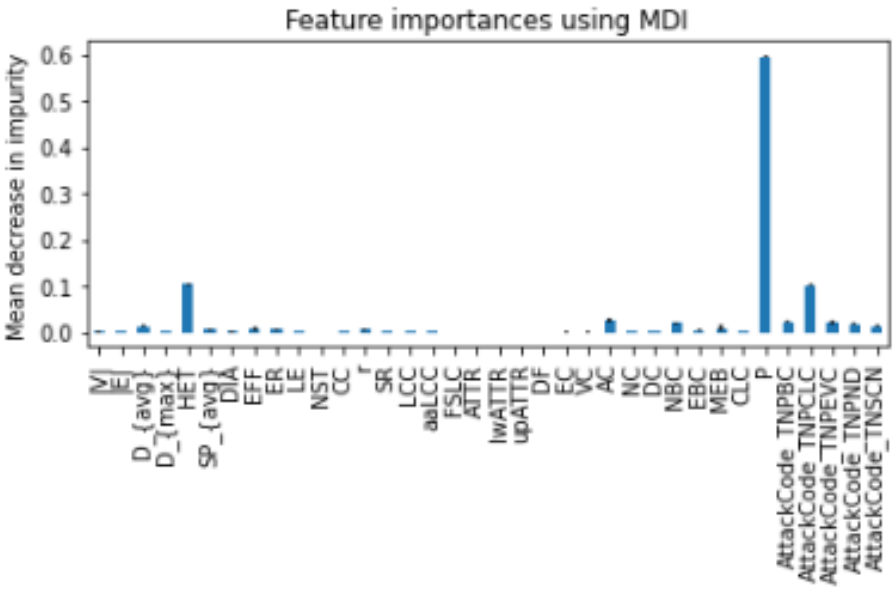

Figura 2: Grafic de barres a l'eix X les mètriques del *dataset* i a l'eix Y la importància relativa sobre la robustesa. Usant un model conjunt pels *datasets* d'algorismes d'atac dirigit.

L'última millora aconsegueix passar de varis models, un per cada tipus d'atac, a dos models definitius: el model per atacs aleatoris i el model per atacs dirigits.

5 Resultats definitius

La precissió dels models definitius es pot veure a la taula 2. En la primera fila hi ha el model per atacs dirigits i en la segona el model per atacs aleatoris.

Dataset	**Precisio**
TanyNSM100	98.65%
RandomNSM100	97.58%

Taula 2: Taula amb els resultats del model a la millora 3.2.

Pel que respecte al cost temporal, a la taula 3, hi ha el temps de càlcul dels diversos *datasets* usats on es pot veure que els càlculs poden tardar varis dies en completar-se, a la figura 3 veiem el resultat d'executar el programa que fa d'interfície on el temps de còmput és de l'odre de **pocs segons**.

Dataset	Temps d'execució
TbcNSM100	29h 06m 52s
TccNSM100	44h 56m 52s
TecNSM100	44h 45m 5s
TndNSM100	44h 42m 56s
TcnNSM100	49h 39m 52s
RandomNSM100	10h 12m 38s

Taula 3: Taula amb els temps d'execució dels diferents datasets usant l'NRS.

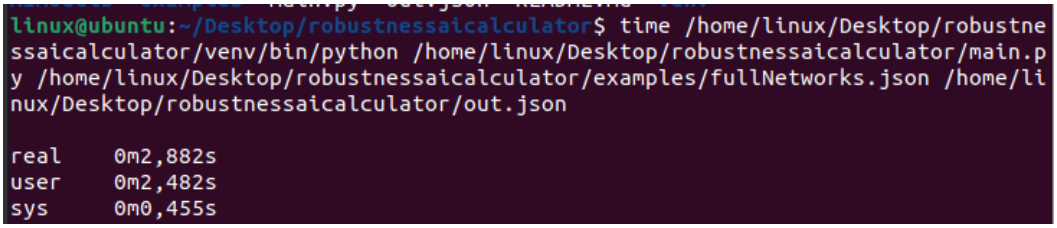

Figura 3: Captura de l'execució i temps de còmput de la interfície de prediccions amb P=[0,30], totes les xarxes i tots els tipus d'atac

6 Conclusions

S'ha aconseguit crear un sistema capaç de calcular la robustesa d'una xarxa de manera instantània. Aquesta eina permet ser-ne la base d'altres com podria ser un recomanador de millores d'una xarxa.

L'eina creada presenta un encert en les prediccions molt alt, per sobre del 97% fet que podria fer-nos plantejar utilitzar aquest model com a substitut del sistema de càlcul actual.

Alguns dels intents d'optimització dels models s'ha hagut de descartar pel fet que empitjoraven els resultats dels models originals.

També s'ha aconseguit detallar les mètriques més rellevants, les que expliquen millor la robustesa d'una xarxa en funció de cada atac. Això permet saber quines mètriques cal reforçar en cas de voler protegir una xarxa davant d'atacs concrets.

Finalment s'ha aconseguit crear una interfície que permet treballar amb els models exportats de manera senzilla i que permetrà, ja sigui a través de l'NRS o de manera individual, fer prediccions sense haver de fer adaptacions al codi ja existent.

Referències

[Marzo 2018] Jose L Marzo, Eusebi Calle, Sergio G Cosgaya, Diego Rueda and Andreu Mañosa. *On selecting the relevant metrics of network robustness*. In 2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM), pages 1–7. IEEE, 2018. (Cited on page .)

[Marzo 2019] Jose L Marzo, Sergio G Cosgaya, Nina Skorin-Kapov, Caterina Scoglio and Heman Shakeri. *A study of the robustness of optical networks under massive failures*. Optical Switching and Networking, vol. 31, pages 1–7, 2019. (Cited on page .)

[Pedregosa 2011] F Pedregosa, G Varoquaux, A Gramfort, B Michel V., Thirion, O Grisel, M Blondel, R Prettenhofer P., Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot and E Duchesnay. *Scikit-learn: Machine Learning in Python*. volume 12, pages 2825–2830, 2011. (Cited on page .)

[Trajanovski 2013] Stojan Trajanovski, Javier Martín-Hernández, Wynand Winterbach and Piet Van Mieghem. *Robustness envelopes of networks*. Journal of Complex Networks, vol. 1, no. 1, pages 44–62, 2013. (Cited on page .)