Redes neuronales como herramienta de asesoramiento en la ciberatribución

Mg Lic Claudio Lopez

Universidad de la Defensa Nacional, Argentina

Abstract.

La Ciberatribución es una parte fundamental de la ciberdefensa de un Estado. La tarea de asignar un responsable de una ciberagresion (y sobre todo si este lo constituye otro Estado) es realmente complicada teniendo en cuenta el avance tecnológico de herramientas afines a los objetivos de los ciberatacantes. Esta actividad (o sea la Ciberatribucion) es fundamental para crear una verdadera disuasión que desaliente la realización de los mencionados ataques.

Por otro lado, la ciencia informática ha desarrollado de manera vertiginosa, la teoría y el empleo de cientos de herramientas de inteligencia artificial que se ven en nuestra vida diaria, y en las que se usan redes neuronales. Las redes neuronales se nutren de miles de datos y sus resultados son más que aceptables para la optimización, clasificación o predicción.

Las fuentes de ingreso de esos datos pueden ser totalmente variadas y de acuerdo con su cantidad se puede afirmar que se logrará mayor o menor precisión en la salida.

Se propone demostrar que las redes neuronales en el contexto de los procedimientos de la Ciberatribucion pueden ser empleadas con éxito como una herramienta de asesoramiento en la determinación del origen de un ataque cibernético a una infraestructura que posea una función crítica.

Si bien existen estudios referidos a este tema, la particularidad de este trabajo es la de inscribirse en el ámbito de ciberdefensa propio en un todo de acuerdo con la legislación nacional vigente
.

Keywords: Ciberdefensa, Ciberdisuacion, aprendizaje automático, redes neurales

1. Introduccion : El problema de la ciberatribucion

El Gen Larry Welch expresa que "... el ciberespacio es un dominio en, desde y a través del cual las operaciones militares producen los efectos deseados. Los objetivos militares fundamentales relativos a este dominio son esencialmente los mismos que en los otros dominios. El objetivo principal es la libertad de acción en, a través y desde el ciberespacio según sea necesario, para apoyar los objetivos de la misión." (Welch, 2011, p. 2) [5].

Para mantener la libertad de acción será necesario, entre otras cuestiones, disuadir al posible adversario de ejecutar agresiones dentro del ciberespacio. Por lo tanto, el objetivo de esa "Ciberdisuacion" será," lograr que los estados naciones agresores, reales o potenciales, perciban claramente que los costos esperados (economics, políticos, militares, geopolíticos, de imagen,

etc.) asociados a una ciberagresión, superan ampliamente a los resultados esperados de la misma" (Uzal, 2016, p. 8) [3].

Ante un Ciberataque, se hace necesario realizar una correcta determinación del origen de este a fin de determinar si el mismo se encuadra dentro de los parámetros del artículo 51 de la Carta de las Naciones Unidas que trata sobre el derecho inmanente de legítima defensa, individual o colectiva en caso de ataque armado contra un Miembro de la ONU [6].

Dado que:

a. Las armas cibernéticas a menudo se despliegan bajo un manto de anonimato, lo que dificulta averiguar quién es realmente responsable
b. Pueden ser desplegadas de manera remota empleando lugares privados o públicos y desde cualquier sitio en el mundo
c. Uno de los escenarios potenciales más desfavorables que se le puede presentar a un estado nación es recibir ciberagresiones y, por incapacidad tecnológica y/o de gestión, terminar adjudicando los desastres ocasionados por dichos ataques a accidentes imprevistos. (Uzal, 2016, p. 9) [3].
d. Cuanto más demoremos en determinar quién es el agresor, el mismo podrá eludir la acción de respuesta (Uzal, 2015, pp. 2-9) [4].

Entonces, la tarea de establecer el responsable último de estas acciones o atribuir o llevar a cabo la "Ciberatribucion" conlleva la dificultad implícita de que ésta deberá tener una alta probabilidad de éxito, a fin de que la respuesta a una ciberagresión pueda ser interpretada como legítima defensa y quedar incluida en los términos del Artículo 51(Uzal, 2015, pp. 2-9) [4].

En el plano interno, recordemos que nuestra Ley de Defensa Nacional y sus Decretos modificatorios (Ley 23.554, Decreto 727/2006 y Decreto 571/2020) expresan en su Artículo 1º "Las Fuerzas Armadas, instrumento militar de la defensa nacional, serán empleadas ante agresiones de origen externo perpetradas por fuerzas armadas pertenecientes a otro/s Estado/s…", por lo tanto, se hace necesario que la ciberatribucion obtenga resultados que tengan en cuenta este aspecto [8].

El Decreto 2645/2014, Directiva de Política de Defensa Nacional, de fecha 30/12/2014, en uno de sus enunciados dice: "…Dentro de la amplia gama de operaciones cibernéticas, sólo una porción de éstas afecta específicamente el ámbito de la Defensa Nacional. En efecto, en materia de ciberdefensa existen dificultades fácticas manifiestas para determinar a priori y ab initio si la afectación se trata de una agresión militar estatal externa. Por tal motivo, resulta necesario establecer dicha calificación a posteriori actuando como respuesta inmediata el Sistema de Defensa únicamente en aquellos casos que se persiguieron objetivos bajo protección de dicho sistema, es decir que poseen la intención de alterar e impedir el funcionamiento de sus capacidades" [7].

Se puede extraer como conclusión entonces que:

- El sistema de Defensa Nacional solo reaccionará ante una agresión (ciberagresion) militar estatal externa y actuará únicamente en aquellos casos en que se afecten infraestructuras bajo su protección.

- En segundo término, la determinación del origen del ataque se hará con posterioridad, es decir, después de un análisis de ciberatribución que puede llevar tanto tiempo que, mientras se realiza, se podrán seguir sufriendo más ataques.

2. Objetivo

El objetivo de esta propuesta será definir un algoritmo empleando redes neuronales, que posibilite dar un adecuado asesoramiento sobre el origen de un ciberataque a fin de contribuir a tomar correctas decisiones por parte del órgano de la ciberdefensa nacional.

3. Desarrollo de la propuesta

3.1. **Metodología**

Para el desarrollo de la propuesta:

- Se empleo un modelo de aprendizaje automático supervisado.
- Se estudió la estructura de datos más conveniente para los valores de entrada. Estos valores debieron ser codificados para el correcto empleo de la red.
- Se tomaron 4 (cuatro) países simulados como valores de etiqueta para cada instancia de vectores de datos tanto de entrenamiento como de validación.
- Se construyó la red neuronal con una cierta cantidad inicial de neuronas y capas, así como cuantificadores específicos que se fueron probado y modificando hasta alcanzar un valor de evaluación aceptable.
- Como métricas cuantitativas se usaron herramientas propias del entorno de programación (ejemplo matriz de confusión). En cuanto a las métricas cualitativas se emplearon casos de atribuciones resueltas y el asesoramiento de expertos.
- Se empleó el entorno de programación abierto Colaboratory de Google,
- Se usó el lenguaje Tensorflow versión 1.14, basado en Python. El mismo constituye una biblioteca de software de código abierto para aprendizaje automático que fue desarrollado por Google a fin de satisfacer las necesidades de sus sistemas

3.2. **Modelado de la propuesta**

Para el conjunto de datos del modelo se eligió una estrategia inicial de muestreo sobre pocas características que podian tener un poder predictivo fuerte. Esto ayudo a confirmar que el modelo funcionara según lo previsto.

Se empleó como herramienta principal el sitio https://www.kaggle.com/. En el se encontraron distintos Datasets vinculados a la ciberseguridad y a la ciberdefensa que permitieron extraer información de referencia. También se usaron datos e información de las siguientes publicaciones:

- NATIONAL CYBERSECURITY AND CYBERDEFENSE POLICY SNAPSHOTS https://observatoriociberseguridad.org/#/home - Reporte de Ciberseguridad para America Latina y el Caribe 2020 [10]
- THE GLOBAL RISKS REPORT 2021 16TH EDITION – Publicación de WORLD ECONOMIC FÓRUM [11].
- Cyber Warfare Conflict Analysis and Case Studies (2017) - Mohan B. Gazula https://ccdcoe.org/library/strategy-and-governance/ [2].
- A Guide to Cyber Attribution"(2018), OFFICE OF THE DIRECTOR OF NATIONAL INTELLIGENCE, US [1]
- MITRE ATT&CK MATRIX FOR ENTERPRISE (2022) [9]

Cada vector de datos representó un determinado caso de ciberataque simulado atribuido a un potencial adversario también simulado.

Las características del modelo son:

- Tipo_ataque
- Objetivo_material
- Efecto_buscado
- Carácter_de_la_agresion
- Confiabilidad_informacion
- Hecho_politico_cercano
- Complejidad_agresion
- Frecuencia_ataque
- Participación_terceros

Como se puede visualizar los aspectos técnicos (como IP de origen, IP de destino, numero de paquetes, etc.) de la ciberagresion no han sido tenidos en cuenta ya que el propósito del modelo será ofrecer un asesoramiento inicial, acerca de las probabilidades atribuidas a cada actor como potencial ejecutor de una agresión.

Fue asignado a cada atributo un rango de valores discretos a fin de facilitar el aprendizaje del modelo.

Todos estos atributos y sus valores han sido analizados y determinados por el autor. Esto no quiere decir que no puedan existir otros. Lo mismo cabe acotar para los atributos.

Estos valores fueron codificarlos a los fines de que el modelo pueda ser ejecutado en el entorno de programación correspondiente. Se emplearon valores numéricos enteros, teniendo como norma establecer un valor 0 cuando el atributo dentro del vector de datos tiene un valor desconocido. La escala de valores se iniciará en 1 (uno) como valor más bajo del rango.

En la siguiente tabla se muestra la asignación de los valores

Tabla 1. Asignación de valores numéricos a los atributos.

Atributo	Valor	Representación en el modelo	Observaciones
Tipo_ataque	Reconocimiento	1	
	Desarrollo de recursos	2	
	Acceso inicial	3	
	Ejecución	4	
	Persistencia	5	
	Descubrimiento de credenciales de acceso	6	
	Movimiento lateral	7	
	Recopilación	8	
	Comando y Control	9	
	Exfiltración	10	
	Impacto	11	
	No hay datos	0	
Objetivo_material	Personas	1	
	Infraestructura critica	2	
	Sistemas	3	
	No hay datos	0	
Efecto_buscado	Robo de información confidencial	1	
	Modificación de archivos	2	
	Negación del servicio	3	
	Destrucción del Sistema Informático	4	
	Sabotaje	5	
	No hay datos	0	
Motivación	Terrorismo	1	
	Espionaje	2	
	Política	3	
	Económica	4	
	No hay datos	0	

Atributo	Valor	Representación en el modelo	Observaciones
Carácter_de_la_agresion	Promovida por un estado	1	
	Ejecutada por un estado haciendo uso de sus FFAA	2	
	Ejecutada por un estado sin hacer uso de sus FFAA	3	
	No hay datos	0	
Confiabilidad_ informacion	Muy Confiable	1	
	Confiable	2	
	No hay datos	0	
Hecho_politico_cercano	Si existió un hecho político cercano	1	
	No existió un hecho político cercano	2	
	No hay datos	0	
Complejidad_agresion	El agresor posee muy buena infraestructura y conocimientos técnicos para la ejecución del ataque	1	
	El agresor posee buena infraestructura y conocimientos técnicos para la ejecución del ataque	2	
	El agresor posee escasa infraestructura y conocimientos técnicos para la ejecución del ataque	3	
	No hay datos	0	
Frecuencia_ataque	Muy frecuente	1	
	Frecuente	2	
	Poco frecuente	0	
Participación_terceros	Hubo participación de terceros	1	
	No hubo participación de terceros	2	
	No hay datos	0	

Se tomaron 5 (cinco) etiquetas para clasificar los datos que representan a los potenciales agresores simulados y se identificaron de acuerdo ala Tabla 2.

En total el Dataset en el inicio contenía 1309 registros de los cuales el 60% (786) fueron empleados como datos de entrenamiento del modelo y el resto (523) como datos de validación. El criterio para la asignación de las etiquetas a cada vector de datos se realizó inicialmente de manera aleatoria a fin de evitar sesgos y suponiendo de que no se hallaron patrones de comportamiento que induzca a que siempre que se presente una determinada característica la etiqueta corresponderá a un Estado particular

Tabla 2. Valores de etiquetas

Etiqueta	Correspondencia
1	ESTADO A
2	ESTADO B
3	ESTADO C
4	ESTADO D
0	INDETERMINADO

3.3. Red neuronal propuesta

En el modelo presentado se usó la clasificación de clase o etiqueta múltiple. Se empleo el criterio de regresión logística. La misma produce un decimal entre 0 y 1.0 en la salida de la red.

Se utilizó como función de activación, la función “sigmoid”, que en redes neuronales es usada para modelos de regresión logística.

Se implemento como función de salida la función Softmax o función exponencial normalizada, que extiende la idea de la regresión logística a un mundo de múltiples clases. Softmax asigna probabilidades decimales a cada clase. Se muestra la red de manera gráfica y esquemática, con la capa de entrada, las capas ocultas y las de salida.

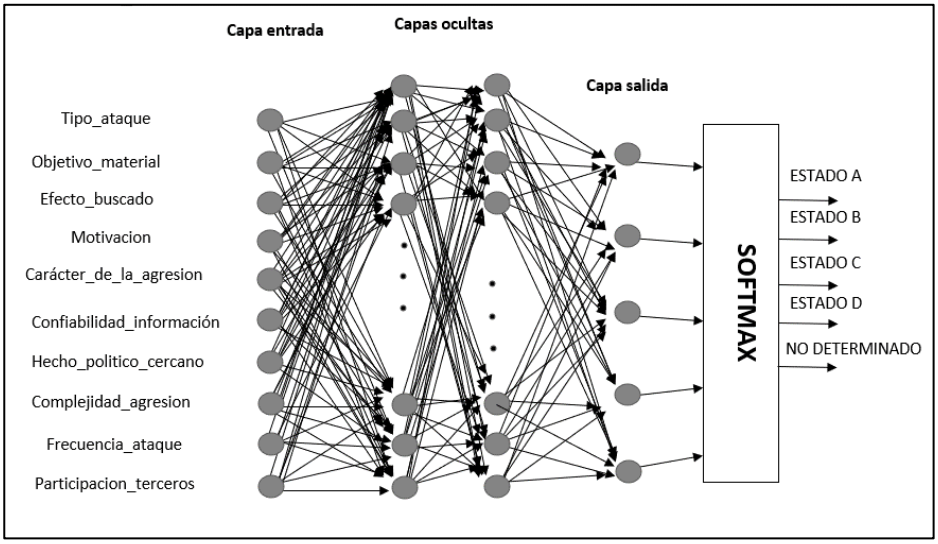


Fig. 1. Esquema de red neuronal

```
model = tf.keras.Sequential()
model.add(keras.layers.Dense(10, activation='sigmoid', input_shape=(10, )))
model.add(keras.layers.Dense(20, activation='sigmoid'))
model.add(keras.layers.Dense(5, activation='softmax'))

model.compile(optimizer="adam",loss="categorical_crossentropy",metrics = ['accuracy'])
```

```
history = model.fit(xs_train,ys_train, epochs=300)
```

Fig.2. Construcción con código de la red

Se muestra en la Figura 2 el código en tensorflow de la construcción de la red empleando la clase keras. En cuanto a los argumentos de optimización y perdida fueron empleados “adam” y “categorical_crossentropy” respectivamente, adam es un método de descenso de gradiente estocástico que se basa en la estimación adaptativa de momentos de primer y segundo orden, categorical _crossentropy se utiliza para el modelo de clasificación de clases múltiples donde hay dos o más etiquetas de salida. La etiqueta de salida, si está presente en forma de número entero, se convierte en codificación categórica mediante keras. Y finalmente se tiene accuracy como métrica que se emplea para monitorizar el proceso de aprendizaje (y prueba) de la red neuronal. Una vez definido el modelo y configurado su método de aprendizaje se invoca al método fit

```
model.summary()
```

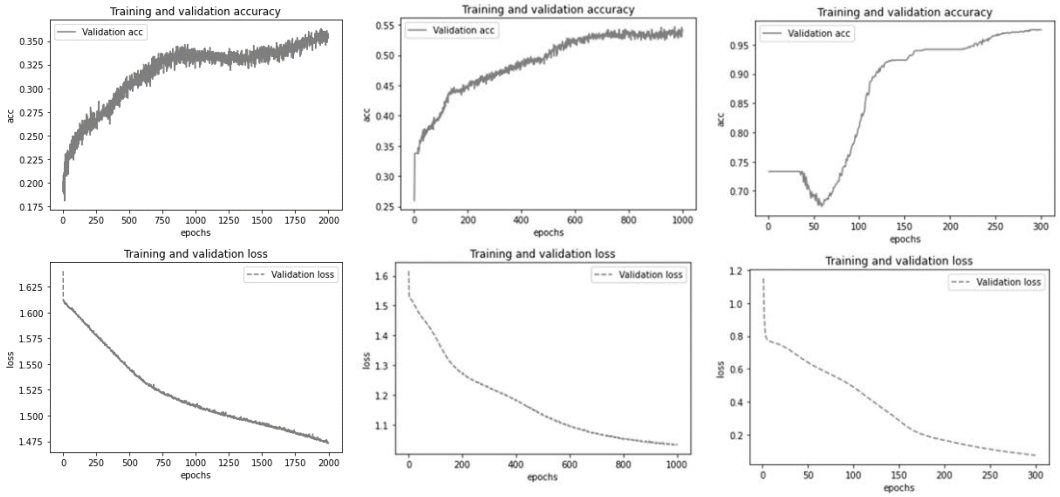
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 10)                110

 dense_4 (Dense)             (None, 20)                220

 dense_5 (Dense)             (None, 5)                 105

=================================================================
Total params: 435
Trainable params: 435
Non-trainable params: 0
_________________________________________________________________
```

Fig.3. Arquitectura de la red

Se observa en la Figura 3 que se requieren 435 parámetros (columna Param #), que corresponden a los 110 parámetros para la primera capa, 220 para la segunda y 105 para la tercera (salida)

3.4. Resultados

Se procedió a ejecutar el modelo con distintas iteraciones o epoch . Los resultados, de manera gráfica, fueron los que muestra la Figura 4**.**



(a) (b) (c)

Fig 4: Resultados ciclos de iteraciones

```
test_loss, test_acc = model.evaluate(xs_test, ys_test)

12/12 [==============================] - 0s 2ms/step - loss: 1.7106 - accuracy: 0.1750

print('Test accuracy:', test_acc)

Test accuracy: 0.17499999701976776
```

Fig. 5: Perdida y precisión de datos de validación 1º iteracion

Si se evalúa el modelo con los datos de validación (“test”) como muestra la Figura 5, o sea con datos nunca vistos por este, en el ciclo de iteraciones de la Figura 4 (a) , se ve que el mismo tiene una precisión del 17 %, Es decir, el modelo es incapaz de generalizar.

Este dato indica que el modelo “no ha aprendido” y que se encontró ante una situación de “overfitting” o sobreaprendizaje

Se dedujo entonces que los conjuntos de datos de entrenamiento y validación podrían no estar correctamente construidos y se revisó la estrategia de separación de estos, hallándose una falta de correspondencia entre ambos que se apreció como una de las causales del problema. En otras

palabras, los datos fueron separados solo a través de un simple corte en la lista que posibilitaba la relación 60 % - 40 % entre tipos de conjunto de datos lo que podía indicar que ambos no eran de la misma “naturaleza”.

A continuación, se elaboró otro conjunto de datos de validación teniendo en cuenta este detalle.

Posteriormente se volvió a ejecutar el modelo, pero esta vez en 1000 iteraciones a fin de detectar rápidamente cualquier problema. Se obtuvieron resultados parecidos (Figura 4 (b)).

Se mejoró la precisión (55% con datos de entrenamiento y 25% con datos de validación) pero se empeoró la perdida 3,12 con datos de validación. (Figura 6)

Se volvió a revisar los conjuntos de datos y se buscó que haya alguna correspondencia en la distribución de estos y sus etiquetas en vez de incrementar la cantidad de vectores de datos. De hecho, se redujo el Dataset a 948 registros. Asimismo, se cambió la relación de los conjuntos de datos haciéndola de 70% - 30%.

Como se aprecia en la Figura 4 (c) el cambio ha sido muy grande lográndose llevar la precisión arriba del 90 % con solo 300 iteraciones. Por lo tanto, se deduce que este sería el modelo más apto a los fines que se buscan.(Figura 7)

```
test_loss, test_acc = model.evaluate(xs_test, ys_test)
```

```
12/12 [==============================] - 0s 2ms/step - loss: 3.1287 - accuracy: 0.2556
```

```
print('Test accuracy:', test_acc)
```

```
Test accuracy: 0.25555557012557983
```

Fig. 6: Perdida y precisión de datos de validación 2º iteracion

```
test_loss, test_acc = model.evaluate(xs_test, ys_test)
```

```
11/11 [==============================] - 0s 3ms/step - loss: 0.1082 - accuracy: 0.9634
```

```
print('Test accuracy:', test_acc)
```

```
Test accuracy: 0.9634146094322205
```

Fig. 7: Perdida y precisión de datos de validación 3º iteracion

Se muestra, además, el resultado total de la clasificación del modelo por medio del método model.predict , y un ejemplo de clasificación en la fila 11 del conjunto de validación donde la predicción se percibe en el valor más alto de los indicados allí , en este caso sería el primero empezando por la izquierda correspondiente al valor 0 o sea país INDETERMINADO.

```
predictions= model.predict(xs_test)
print(predictions)
```

```
[[9.37825501e-01 5.70566058e-02 1.94890879e-03 3.45552166e-04
  2.82340520e-03]
 [9.33681071e-01 6.03133403e-02 2.79712165e-03 1.98428682e-03
  1.22422550e-03]
 [8.71142805e-01 7.80616179e-02 1.20318625e-02 5.89643923e-06
  3.87577079e-02]
 ...
 [4.56051528e-01 1.30183995e-01 3.38017778e-03 1.50364201e-06
  4.10382777e-01]
 [4.52811569e-01 1.31552175e-01 3.46387923e-03 1.51578649e-06
  4.12170827e-01]
 [4.49889928e-01 1.33059859e-01 3.56233446e-03 1.53056590e-06
  4.13486332e-01]]
```

```
print(predictions[11])
```

```
[9.8961020e-01 4.8194937e-03 1.2330770e-03 4.3200920e-03 1.7179977e-05]
```

Fig 8: Valores de predicción del modelo

Se han empleado herramientas para la evaluación del modelo como lo es la matriz de confusión o confusión_matrix. La misma constituye una tabla con filas y columnas que contabilizan las predicciones en comparación con los valores reales

En la Figura 9 se observa aplicada en el modelo

```
confusion_matrix(ys_test.argmax(axis=1), predictions.argmax(axis=1))
```

```
array([[207,   0,   0,   0,   0],
       [  8,   0,   0,   0,   0],
       [  0,   0,  94,   0,   0],
       [  0,   0,   0,  15,   0],
       [  4,   0,   0,   0,   0]])
```

Fig 9: Matriz de confusión del modelo

Si se desea probar un vector cualquiera se puede apreciar en la Figura 13 que el resultado predicho por la red corresponde al Estado B o sea el que estaría involucrado en el ataque.

3.5. Discusión

Una de las observaciones que se pueden hacer de la construcción de la red es que el principal problema no fue el código que se elaboró sino el tratamiento que se le dio a los datos.

Los resultados fueron más precisos cuando, en primer lugar, se realizó una distribución más acorde entre los conjuntos de datos de entrenamiento y validación y posteriormente se estableció alguna relación entre las características y etiquetas, o sea cuando el modelo "encontró una conexión" entre ambos.

Sin embargo, se deberá tener cuidado con esto último ya que podrá llevar a un riesgo de sesgamiento de los datos, por lo que corresponderá analizar la estrategia de construcción de los Dataset de entrenamiento y validación y evitar el overfitting.

Una de las soluciones para atender este problema sería aumentar sustancialmente la cantidad de datos involucrados en el modelo, cuestión que no asegura el éxito.

Asimismo, como se sabe, el entorno legal donde trabajaría el modelo nos obligaría a tener en cuenta sólo aquellos ciberataques atribuidos a un adversario externo, estatal y militar.

4. Conclusiones

La complejidad del ciberespacio hace muy difícil suponer que se está totalmente blindado contra las intrusiones maliciosas en los sistemas. Por lo tanto, se debe hacer la suposición de que es muy probable que una infraestructura critica propia esté ya siendo víctima, por diversas motivaciones, de un ciberataque perpetrado por hackers apoyados por algún Estado.

La Ciberdisuacion hoy se convierte en un eje relevante en la ciberdefensa de cualquier Estado-Nacion. La "amenaza" de las consecuencias que puede tener cualquier acción contra las infraestructuras criticas nacionales llega a ser un factor clave.

Esto se fortifica si se está en capacidad de lograr certeza en la ciberatribución, por ello contar con instrumentos eficaces como una inteligencia de amenazas adecuada, tecnologías modernas como la Inteligencia artificial, legislación que tenga en cuenta las características estratégicas de los tiempos actuales y otros, será imprescindible para lograr "identificar" a los responsables de la agresión.

La solución propuesta pudo beneficiarse con una de las herramientas de la inteligencia artificial como lo son las redes neuronales y obtener resultados que pueden llegar a ser muy útiles como ayuda a la decisión.

El modelo no es complicado de emplear, es altamente escalable, portable y su mantenimiento (actualización) es sencillo.

Poniendo foco en los datos nuevamente, su análisis y ponderación se hacen fundamentales para su incorporación al modelo.

Su ámbito de aplicación sería un Comando Conjunto de la Ciberdefensa, y su función principal la de constituirse en una herramienta más de asesoramiento para la determinación del origen de un ciberataque a infraestructuras críticas propias.

La información volcada en él será la que ese Comando Conjunto considere como valida y útil para que las salidas sean aprovechables.

5. Referencias.

1. "A Guide to Cyber Attribution"(2018), OFFICE OF THE DIRECTOR OF NATIONAL INTELLIGENCE US, https://www.dni.gov/files/CTIIC/documents/ODNI_A_Guide_to_Cyber_Attribution.pdf
2. Carta de las Naciones Unidas
3. Decreto 2645/2014 - DIRECTIVA DE POLÍTICA DE DEFENSA NACIONAL – ACTUALIZACIÓN
4. Gazula Mohan B. , (2017), Cyber Warfare Conflict Analysis and Case Studies, Cybersecurity Interdisciplinary Systems Laboratory (CISL)- Sloan School of Management, Room E62-422 - Massachusetts Institute of Technology.
5. Ley 23.554 de DEFENSA NACIONAL
6. MITRE ATT&CK MATRIX FOR ENTERPRISE (2022). https://attack.mitre.org/matrices/enterprise/
7. NATIONAL CYBERSECURITY AND CYBERDEFENSE POLICY SNAPSHOTS
8. THE GLOBAL RISKS REPORT 2021 16TH EDITION – Publicación de WORLD ECONOMIC FÓRUM
9. Uzal Roberto, (2016) Ciber Disuasión. Un capítulo particularmente sensitivo de la Ciberdefensa, BOLETÍN DEL ISIAE Instituto de Seguridad Internacional y Asuntos Estratégicos Número 64. 8-18
10. Uzal Roberto (2015), El Problema de la Ciber Atribución: Aportes para una estrategia de Ciber Defensa BOLETÍN DEL ISIAE Número 61, 2 – 9
11. Welch Larry D. Usaf (Ret.), s.f. "Cyberspace – the Fifth Operational Domain". Institute for Defense Analysis https://www.ida.org/-/media/feature/publications/2/20/2011-cyberspace---the-fifth - operational-domain/2011-cyberspace---the-fifth-operational-domain.ashx