UNIVERSIDADE DA CORUÑA

**Escuela Internacional de Doctorado de la Universidade da Coruña (EIDUDC)**

*Programa Oficial de Doctorado en Energía y Propulsión Marina*

Tesis Doctoral

# *Análisis y detección de ataques informáticos mediante sistemas inteligentes de reducción dimensional*

**Autor:** Rafael Alejandro Vega Vega

**Directores**: Héctor Quintián Pardo
Esteban Jove Pérez

Ferrol, Enero de 2022

UNIVERSIDADE DA CORUÑA

Dr. Héctor Quintián Pardo y Dr. Esteban Jove Pérez, profesores del Departamento de Ingeniería Industrial de la Universidade da Coruña,

AUTORIZAN

A la defensa de la Tesis Doctoral titulada *"Análisis y detección de ataques informáticos mediante sistemas inteligentes de reducción dimensional"*, realizada por D. Rafael Alejandro Vega Vega, bajo nuestra dirección y supervisión, y que presenta para la obtención del grado de Doctor por la Universidade da Coruña.

En Ferrol, a 1 de Febrero de 2022

Dr. Héctor Quintián Pardo

Dr. Esteban Jove Pérez

*A mi mujer Mari Carmen, por compartir mi vida.*
*A mis hijas Marina y Emma.*
*Gracias por vuestro cariño y apoyo.*

*A mis padres, por ayudarme a ser lo que soy.*

*A mis queridos cuñados Andrés y Agustín, in memoriam.*

# Agradecimientos

Quisiera expresar aquí mi más profundo agradecimiento a aquellas personas que han hecho posible esta tesis doctoral.

A mis directores de tesis Héctor Quintián Pardo y Esteban Jove Pérez por su gran esfuerzo y dedicación, ya que gracias a ellos ha sido posible la realización de esta tesis.

A José Luis Calvo Rolle por el enorme trabajo realizado.

A José Luis Casteleiro Roca, por su ayuda constante en la edición del documento.

Al Departamento de Ingeniería Industrial por apoyarme en la realización de esta Tesis Doctoral.

*Rafael Alejandro Vega Vega*
*Ferrol, Enero de 2022*

# Resumen

El presente trabajo de investigación aborda el estudio y desarrollo de una metodología para la detección de ataques informáticos mediante el uso de sistemas y técnicas inteligentes de reducción dimensional en el ámbito de la ciberseguridad. Con esta propuesta se pretende dividir el problema en dos fases. La primera consiste en un reducción dimensional del espacio de entrada original, proyectando los datos sobre un espacio de salida de menor dimensión mediante transformaciones lineales y/o no lineales que permiten obtener una mejor visualización de la estructura interna del conjunto de datos. En la segunda fase se introduce el conocimiento de un experto humano que permite aportar su conocimiento mediante el etiquetado de las muestras en base a las proyecciones obtenidas y su experiencia sobre el problema. Esta novedosa propuesta pone a disposición del usuario final una herramienta sencilla y proporciona unos resultados intuitivos y fácilmente interpretables, permitiendo hacer frente a nuevas amenazas a las que el usuario no se haya visto expuesto, obteniendo resultados altamente satisfactorios en todos los casos reales en los que se ha aplicado.

El sistema desarrollado ha sido validado sobre tres supuestos reales diferentes, en los que se ha avanzado en términos de conocimiento con un claro hilo conductor de progreso positivo de la propuesta. En el primero de los casos se efectúa un análisis de un conocido conjunto de datos de *malware* de Android en el que, mediante técnicas clásicas de reducción dimensional, se efectúa una caracterización de las diversas familias de *malware*. Para la segunda de las propuestas se trabaja sobre el mismo conjunto de datos, pero en este caso se aplican técnicas más avanzadas e incipientes de reducción dimensional y visualización, consiguiendo que los resultados se mejoren significativamente. En el último de los trabajos se aprovecha el conocimiento de los dos trabajos previos, y se aplica a la detección de intrusión en sistemas informáticos sobre datos de redes, en las que se producen ataques de diversa índole durante procesos de funcionamiento normal de la red.

**Palabras clave:** *Reducción Dimensional, Proyección Visual, Inteligencia Artificial, Redes Neuronales, Aprendizaje no Supervisado, Ciberseguridad, Malware*

# Abstract

This research work addresses the study and development of a methodology for the detection of computer attacks using intelligent systems and techniques for dimensional reduction in the field of cybersecurity. This proposal is intended to divide the problem into two phases. The first consists of a dimensional reduction of the original input space, projecting the data onto a lower-dimensional output space using linear or non-linear transformations that allow a better visualization of the internal structure of the dataset. In the second phase, the experience of an human expert is presented, which makes it possible to contribute his knowledge by labeling the samples based on the projections obtained and his experience on the problem. This innovative proposal makes a simple tool available to the end user and provides intuitive and easily interpretable results, allowing to face new threats to which the user has not been exposed, obtaining highly satisfactory results in all real cases in which has been applied.

The developed system has been validated on three different real case studies, in which progress has been made in terms of knowledge with a clear guiding thread of positive progress of the proposal. In the first case, an analysis of a well-known Android malware dataset is carried out, in which a characterization of the various families of malware is developed using classical dimensional reduction techniques. For the second of the proposals, it has been worked on the same data set, but in this case more advanced and incipient techniques of dimensional reduction and visualization are applied, achieving a significant improvement in the results. The last work takes advantage of the knowledge of the two previous works, which is applied to the detection of intrusion in computer systems on network dataset, in which attacks of different kinds occur during normal network operation processes.

**Keywords:** *Dimensional Reduction, Visual Projection, Artificial Intelligence, Neural Networks, Unsupervised Learning, Cybersecurity, Malware*

# Resumo

Este traballo de investigación aborda o estudo e desenvolvemento dunha metodoloxía para a detección de ataques informáticos mediante o uso de sistemas e técnicas intelixentes de redución dimensional no ámbito da ciberseguridade. Esta proposta pretende dividir o problema en dúas fases. A primeira consiste nunha redución dimensional do espazo de entrada orixinal, proxectando os datos nun espazo de saída de menor dimensionalidade mediante transformacións lineais ou non lineais que permitan unha mellor visualización da estrutura interna do conxunto de datos. Na segunda fase, introdúcese a experiencia dun experto humano, que lle permite achegar os seus coñecementos etiquetando as mostras en función das proxeccións obtidas e da súa experiencia sobre o problema. Esta proposta innovadora pon a disposición do usuario final unha ferramenta sinxela e proporciona resultados intuitivos e facilmente interpretables, que permiten facer fronte a novas ameazas ás que o usuario non estivo exposto, obtendo resultados altamente satisfactorios en todos os casos reais nos que se aplicou.

O sistema desenvolvido validouse sobre tres supostos reais diferentes, nos que se avanzou en canto ao coñecemento cun claro fío condutor de avance positivo da proposta. No primeiro caso, realízase unha análise dun coñecido conxunto de datos de malware Android, no que se realiza unha caracterización das distintas familias de malware mediante técnicas clásicas de redución dimensional. Para a segunda das propostas trabállase sobre o mesmo conxunto de datos, pero neste caso aplícanse técnicas máis avanzadas e incipientes de redución dimensional e visualización, conseguindo que os resultados se melloren notablemente. O último dos traballos aproveita o coñecemento dos dous traballos anteriores, e aplícase á detección de intrusos en sistemas informáticos en datos da rede, nos que se producen ataques de diversa índole durante os procesos normais de funcionamento da rede.

**Keywords:** *Redución Dimensional, Proxección Visual, Intelixencia Artificial, Redes Neuronais, Aprendizaxe non Supervisada, Ciberseguridade, Malware*

# Índice general

# Índice de figuras

# Índice de tablas

# CAPÍTULO 1

# Introducción

En este capítulo se exponen, en primer lugar, los antecedentes que dan lugar a la elaboración de este trabajo. Posteriormente, se aborda la motivación que se persigue con esta tesis doctoral, centrándose en los problemas más usuales encontrados, y en la posibilidad de mejorarlos. Los últimos puntos describen tanto los objetivos específicos del trabajo, la metodología utilizada para su consecución, así como la exposición de la estructura principal del documento.

## 1.1. Antecedentes

A lo largo de las últimas décadas, el concepto de ciberseguridad ha ido ganando peso en infinidad de ámbitos, tales como la industria o las telecomunicaciones, entre otros [1]. Se define la ciberseguridad como un conjunto de procesos y tecnologías diseñadas con el objetivo de proteger programas, ordenadores, redes de comunicación y también datos ante ataques, y/o accesos no autorizados, asegurando de esta manera la confidencialidad, integridad y disponibilidad de los sistemas [2]. A pesar de que no es posible garantizar una seguridad total, la ciberseguridad tiene como objetivo evitar los ataques maliciosos, reducir la vulnerabilidad de la información consecuencia de errores propios y paliar los daños ocasionados como consecuencia de los mismos [3]. Dependiendo de sus características, se consideran varios tipos de incidentes a los que un sistema de ciberseguridad debe hacer frente [4]:

- Acceso no autorizado de información de una red, sistema o conjunto de datos.
- *Software* malicioso diseñado para dañar un ordenador, servidor, red, etc.
- Denegación de servicio (DoS) de un sistema, forzando su inutilización.

- El conocido como *phishing*, que se nutre de la interacción fraudulenta de usuarios para obtener información sensible sobre cuentas bancarias, redes sociales, etc.
- Ataque día cero llevado a cabo sobre un aspecto vulnerable no conocido en el sistema de seguridad.

Ante el aumento exponencial de los incidentes de esta naturaleza experimentados a lo largo de los últimos años, se plantean una serie de puntos de partida ineludibles a la hora de implementar un sistema de ciberseguridad [5]:

- Conocer el estado de la red. Incluye dispositivos de interconexión, puertos de transmisión de datos, equipos, etc.
- Conocer el estado de los servicios que se ofrecen.
- Conocer las actividades que se están desarrollando en cada momento.
- Tener una idea de las tendencias que se observan.
- Recibir alarmas ante eventos e incidencias relevantes.
- Almacenar el conjunto de datos en un histórico.
- Prever futuras necesidades/anticipar futuros problemas gracias a los datos de tendencia.

Uno de los ámbitos en los que más relevancia ha ido ganando el concepto de ciberseguridad es el de la industria [6]. Los sistemas industriales clásicos estaban formados por una estructura piramidal con cinco niveles bien diferenciados (figura 1.1) [7]: campo, control, supervisión, planificación y gestión. Si bien existía interconexión entre dos niveles contiguos, esta estructura jerárquica carecía de comunicación entre niveles alejados en la pirámide. De la misma manera, la conexión con el exterior era prácticamente inexistente, haciendo uso de aplicaciones específicas [8].

Sin embargo, la denominada cuarta revolución industrial ha puesto sobre la mesa una evolución de los sistemas industriales [9]. Por una parte, éstos presentan mayor comunicación con el exterior, haciendo uso de aplicaciones más genéricas y *software* libre [10]. Por otra parte, se ha avanzado hacia una topología distribuida, basada en una mayor conectividad y flexibilidad, favoreciendo una mayor adaptabilidad ante la

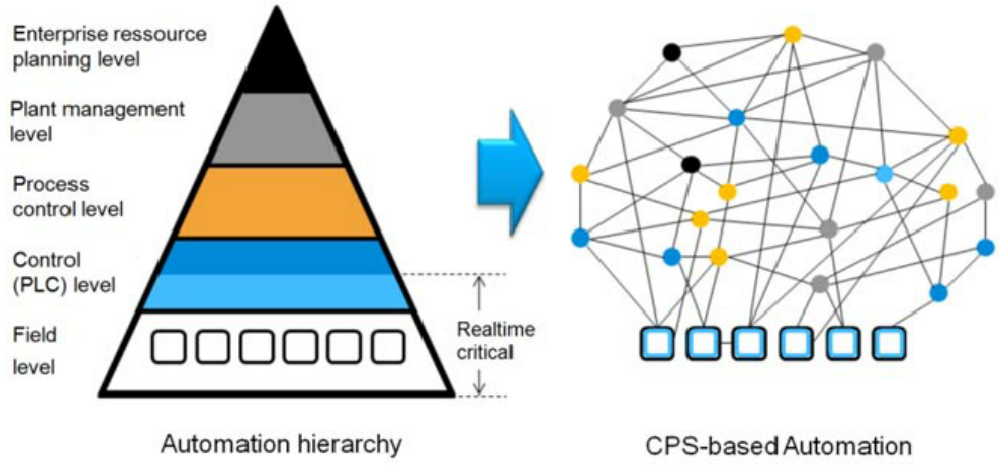


Figura 1.1: Evolución de la estructura de un proceso industrial

demanda de un mercado cada vez más globalizado [11]. En este nuevo paradigma, el empleo de herramientas de digitalización y el *Internet of Things* (IoT) favorecen la comunicación entre los elementos de los distintos niveles, ya sean operarios, sensores o personal de gestión, entre otros [12].

A pesar de los beneficios que implica el incremento de la interconectividad, el mayor flujo de datos supone un aumento del riesgo de sufrir cualquier tipo de ataque que ponga en peligro el correcto funcionamiento de un proceso [13]. Con el fin de ayudar en la toma de decisiones y garantizar la correcta operación de una instalación, velando por la seguridad del mismo, durante la monitorización se han de registrar innumerables variables que entran en juego y que están caracterizadas por lo que se conoce como las '4 v' [14]: velocidad, volumen, variedad y veracidad. Estas características, inherentes a la naturaleza de los datos monitorizados en cualquier proceso industrial, hacen que un observador humano encuentre gran dificultad a la hora de tomar decisiones que favorezcan la ciberseguridad [15].

Ante esta tesitura, se recurre al concepto multidisciplinar conocido como *data science*, que abarca la recopilación de datos, el análisis estadístico y procesado de los mismos, la detección de patrones, extracción de información valiosa y ayuda en la toma de decisiones [16]. Este novedoso concepto, capaz de lidiar con el volumen, velocidad y variedad de los datos, se presenta como una herramienta de gran utilidad en el ámbito de la ciberseguridad [17].

A la hora de emplear técnicas inteligentes, el conocimiento previo acerca del sistema a proteger da lugar a tres tipos de enfoques posibles [18]. Por una parte, se puede considerar la existencia tanto de datos correspondientes al correcto funcionamiento del sistema, como de los eventos anómalos a detectar. En ese caso, el empleo de técnicas de aprendizaje supervisado se aplica para determinar la naturaleza normal o anómala de los datos [19]. Otro posible enfoque comúnmente utilizado se lleva a cabo cuando sólo se dispone de información acerca del funcionamiento habitual del sistema, en cuyo caso se emplean técnicas de tipo semisupervisado, capaces de etiquetar las instancias anómalas aún sin tener previo conocimiento acerca de las mismas [20]. Este enfoque se desarrolla a partir del uso de técnicas de tipo one-class. Finalmente, el empleo de técnicas basadas en aprendizaje no supervisado pretende etiquetar los datos en ausencia de conocimiento previo acerca del comportamiento de los mismos [21].

Son muchos los retos que se plantean en cuanto a ciberseguridad de cara al futuro [22]. La apertura de los sistemas industriales es un aspecto inaplazable, en el que se están introduciendo todas las prerrogativas que concede la informática de consumo. Sin embargo, no se puede dejar a merced de actitudes despiadadas y casi siempre injustificables, las acciones de ataques informáticos, que pudieran recaer en sectores tan críticos y/o estratégicos como pueden ser el nuclear o, desde un punto de vista más general, el energético [23].

La competitividad en el sector industrial indudablemente va a pasar por abandonar la actitud relativamente conservadora que la ha caracterizado en los últimos tiempos, y que era, en cierta medida, un mecanismo de seguridad [24]. Los proveedores de productos industriales ya incorporan en sus catálogos la protección desde un punto de vista de ciberseguridad. Estos servicios han de incluir sistemas de protección escalables, disponiendo de mecanismos de identificación con una robustez infranqueable y adaptable en el tiempo [25]. Por supuesto han de incorporar todas las prestaciones de los diferentes avances que se vayan produciendo, y han de ser lo suficientemente ágiles para progresar de forma pareja a la informática más avanzada del momento [26].

Todo lo anterior va a venir de la mano de implantación de políticas en las que se incorpore la ciberseguridad desde los estados más incipientes [27]. No es una cuestión que pase únicamente por contar con tecnologías avanzadas en este sentido, si no de que éstas estén embebidas en los propios procesos de gestión, sin dejar de lado todos los factores y actores implicados en las instituciones [28]. Con el objetivo de conferir alguna de las características anteriormente mencionadas, es vital, por ejemplo para una actualización constante, el manejo de toda la información, así como su uso [29].

El tratamiento se ha de hacer de una forma inteligente y que permita escalabilidad ante nuevas tecnologías, que con toda seguridad se consolidarán, como puede ser el uso del *cloud* en todos los procesos en que sea posible [30]. Por supuesto, esta afirmación conlleva que la inteligencia de los sistemas, así como el conocimiento del que están provistos, se vaya adaptando a las nuevas exigencias, y que lo haga con la rapidez suficiente que permita garantizar que las instalaciones industriales no sean vulnerables a posibles ataques que se dan o puedan darse en el futuro [31]. Es necesario hacer especial énfasis en esta última parte, dada la celeridad con la que aparecen nuevas amenazas, independientemente de la peligrosidad que puedan entrañar [32].

No cabe duda que para poder efectuar un mantenimiento y prevención óptimo de ataques desde un punto de vista digital es primordial poder contar con conocimientos especializados, tanto técnicos como estratégicos [33]. De esta forma se podrá realizar una evaluación fidedigna del riesgo existente y las metas a perseguir, realizando actualizaciones periódicas con ese fin, tratando de ser y, por supuesto, parecer seguros desde el punto de vista de la ciberseguridad [34].

## 1.2. Motivación

Desde hace ya un tiempo, son muchos los esfuerzos que se vienen realizando en términos de seguridad desde un punto de vista global. En casi todos los casos se ha pretendido en cierta medida una automatización de las diferentes etapas que conlleva este concepto. El motivo subyacente y lógico es que se ha de tener en cuenta que los usuarios de este tipo de sistemas carecen de formación especializada, que contribuya a efectuar configuraciones que conlleven altos requerimientos o niveles de especialización.

Dado que uno de los objetivos es incrementar la precisión en la detección de amenazas para la posterior implementación de la protección, se considera necesario retroceder en cierta medida en cuanto a la creación de métodos, procedimientos o estrategias que permitan llevar a cabo automáticamente las tareas requeridas para disponer de sistemas seguros.

Surge por tanto la principal motivación de esta investigación, con la que se desea disponer de una herramienta lo más sencilla e intuitiva posible, que permita determinar el estado de una red informática ante la posibilidad de que se produzca un eventual ataque. Por supuesto no se debe ver afectada la eficacia, en todo caso mejorada.

Se perseguirán además los siguientes aspectos en la voluntad de la consecución de la anterior motivación:

- Se tratará de mitigar la identificación de falsas amenazas.
- Se apelará a la experiencia y ratificación por parte de un humano.
- Dispondrá de la posibilidad de afrontar nuevos escenarios de amenazas hasta el momento desconocidos.

La motivación de este trabajo de investigación tiene dos vertientes claras. Por un lado, se busca una nueva metodología para la identificación de amenazas desde un punto de vista de seguridad. Y por otra parte, el trabajo tiene el propio interés de la aplicación de la metodología a tres problemas concretos en el marco de la seguridad y en el contexto de un mundo conectado. Se implementará en dos áreas de gran interés en este momento: protección de *Smartgrids* ante vulnerabilidades existentes o posibles ataques, y por otro lado con *malware* de Android, sobre un conocido y estudiado conjunto de datos público, el Malgenome.

## 1.3. Objetivos

El objetivo que se plantea en esta tesis es el diseño de una estrategia de análisis y detección de ataques informáticos, basado en sistemas inteligentes fundamentados en reducción dimensional. La propuesta debe basarse en datos fidedignos en los que haya casos sin presencia de ataques, y otros en los que si existan, y que además estén etiquetados.

Se tratará de disponer de conjuntos de ataques de diversa índole con diferente casuística en términos de análisis de datos, como pueden ser, por ejemplo, sistemas no balanceados. Para la validación de la propuesta se considerará su aplicación para dar respuesta a problemas concretos: análisis y caracterización de familias *malware*, tanto con métodos conocidos como con la aplicación de propuestas incipientes. Por otro lado, se tratarán de identificar ciberataques que producen comportamientos anómalos sobre el manejo de protocolos de redes.

De forma detallada los objetivos específicos planteados son los siguientes:

- Alcanzar una propuesta de una metodología basada en métodos comunes, en la que se efectúa el análisis de una estructura interna de un conjunto de datos con diferentes ataques.
- La metodología propuesta en el punto anterior se tratará de llevar a cabo con métodos avanzados e incipientes con el mismo objetivo, para reafirmar así la validez.
- Por último se tratará de conseguir una propuesta para detectar ciberataques sobre conjuntos de datos, tanto de los que poseen amenazas, como otros que carecen de ellas.

## 1.4. Metodología

La metodología que se ha seguido a lo largo de este trabajo para alcanzar los objetivos propuestos se puede resumir en los siguientes puntos:

- Documentación: Se lleva a cabo una búsqueda y análisis de documentación que permite tener una visión detallada de los antecedentes y estado actual de la materia.
  - Estudio de tipos y principios de funcionamiento de diversas familias de ciberataques.
  - Estudio de los principales algoritmos de aprendizaje no supervisado, aplicados en ciberseguridad.
- Plan de trabajo: Se realiza un plan para realizar el trabajo de investigación en el que se tratan de alcanzar los objetivos de partida. La planificación se puede desglosar en los siguientes puntos principales.
  - Obtención de conjuntos de datos reales sobre ciberataques.
  - Analisis inicial de los datos obtenidos.
  - Aplicación de los algoritmos de reducción dimensional estudiados.
  - Uso de nuevo/s algoritmo/s que mejoren el rendimiento de los usados en el punto anterior.
  - Analisis de los resultados:

  - Comparar el algoritmo propuesto con los estudiados.
  - Validar los resultados y mejoras obtenidas por el algoritmo propuesto bajo casos de estudio reales.
- Exposición de las conclusiones finales.
- Descripción de los posibles trabajos futuros.

## 1.5. Estructura

Debido a que este trabajo se presenta en la modalidad de compendio de publicaciones, la estructura de este documento no guarda relación directa con la metodología mencionada. Después del presente capítulo de Introducción, se presentan los métodos utilizados en la detección de ciberataques. En el capítulo 3, se describe de forma detallada la propuesta presentada en este trabajo, incluyendo el procedimiento para analizar los sistemas informáticos objeto de ciberataque.

A continuación, en los capítulos 4, 5 y 6, se presentan tres artículos con distintos casos de estudios donde se aplicó el sistema desarrollado obteniendo buenos resultados.

En el primer caso de estudio, capítulo 4, se realizó un analisis del conjunto de datos "The Malgenome dataset" [35], proveniente del projecto "Android Malware Genome" [36]. En dicho analisis se realizó una caraterización de las diversas familias de *malware* analizadas mediante el uso de técnicas clasicas de reducción dimensional y *Exploratory Projection Pursuit (EPP)*, como son *Principal Component Analisys (PCA)*, *Maximum Likelihood Hebbian Learning (MLHL)*, *Cooperative Maximum Likelihood Hebbian Learning (CMLHL)*, *Isometric Mapping (ISOMAP)*, *Curvilinear Component Analysis (CCA)*, *Self Organizing Map (SOM)*. Mediante proyecciones visuales (2D) obtenidas por los algoritmos mencionados, se realiza un análisis en base a las características de cada tipo de *malware* tratando de identificar aquellos patrones comunes entre muestras (tipos de *malware*) proyectados en zonas cercanas.

En el segundo artículo, capítulo 5, se usa el conjuto de datos de *malware* anterior, para aplicar un nuevo algoritmo de EPP, comparando los resultados de las agrupaciones obtenidas por los mejores algoritmos del paso anterior, con las nuevas proyecciones obtenidas por *Beta Hebbian Learning (BHL)*, destacando este último sobre los demás con una mejor visualización de la estructura interna del conjunto de datos. Por último

se validan los resultados mediante árboles de decisión con el objetivo de comparar la característica que en BHL se identifica como más relevante con las más elegidas por los árboles de decisión empleados.

En el tercer artículo, correspondiente al capítulo 6, se emplea el conocimiento generado en los primeros artículos con objeto de aplicar la metodología seguida en la detección de intrusión en sistemas informáticos, mediante la proyección visual en 2-3D de flujos de datos en redes informáticas en las que se producen diversos tipo de ataques.

Para finalizar esta tesis, en los capítulos 7 y 8, se exponen las conclusiones y trabajos futuros tanto de manera general como específicos para los casos de estudio mostrados. En la última parte del documento, se incluyen tanto las portadas de los artículos, como un listado de los trabajos de investigación del autor. En el último apartado de este documento se recogen todas las referencias bibliográficas utilizadas a lo largo de la tesis.

## Bibliografía

[1] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014.

[2] D. Craigen, N. Diakun-Thibault, and R. Purse, "Defining cybersecurity," *Technology Innovation Management Review*, vol. 4, no. 10, 2014.

[3] P. W. Singer and A. Friedman, *Cybersecurity: What everyone needs to know*. oup usa, 2014.

[4] M. Uma and G. Padmavathi, "A survey on various cyber attacks and their classification." *Int. J. Netw. Secur.*, vol. 15, no. 5, pp. 390–396, 2013.

[5] A. M. Shabut, K. T. Lwin, and M. A. Hossain, "Cyber attacks, countermeasures, and protection schemes—a state of the art survey," in *2016 10th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. IEEE, 2016, pp. 37–44.

[6] C.-T. Lin, S.-L. Wu, and M.-L. Lee, "Cyber attack and defense on industry control systems," in *2017 IEEE Conference on Dependable and Secure Computing*. IEEE, 2017, pp. 524–526.

[7] C. J. Bartodziej, "The concept industry 4.0," in *The concept industry 4.0*. Springer, 2017, pp. 27–50.

[8] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE industrial electronics magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[9] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business and information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[10] L. Thames and D. Schaefer, "Software-defined cloud manufacturing for industry 4.0," *Procedia CIRP*, vol. 52, pp. 12–17, 2016, the Sixth International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2016). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827116307910

[11] J. C. Bendul and H. Blunck, "The design space of production planning and control for industry 4.0," *Computers in Industry*, vol. 105, pp. 260–272, 2019.

[12] A. Rojko, "Industry 4.0 concept: Background and overview." *International Journal of Interactive Mobile Technologies*, vol. 11, no. 5, 2017.

[13] J. Tupa, J. Simota, and F. Steiner, "Aspects of risk management implementation for industry 4.0," *Procedia manufacturing*, vol. 11, pp. 1223–1230, 2017.

[14] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *Ieee Access*, vol. 6, pp. 3585–3593, 2018.

[15] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, "Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Annals of operations research*, pp. 1–19, 2020.

[16] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *Journal of Big data*, vol. 7, no. 1, pp. 1–29, 2020.

[17] M. Lezzi, M. Lazoi, and A. Corallo, "Cybersecurity for industry 4.0 in the current literature: A reference framework," *Computers in Industry*, vol. 103, pp. 97–110, 2018.

[18] T. Nguyen, R. G. Gosine, and P. Warrian, "A systematic review of big data analytics for oil and gas industry 4.0," *IEEE Access*, vol. 8, pp. 61 183–61 201, 2020.

[19] R. Gupta, S. Tanwar, N. Kumar, and S. Tyagi, "Blockchain-based security attack resilience schemes for autonomous vehicles in industry 4.0: A systematic review," *Computers & Electrical Engineering*, vol. 86, p. 106717, 2020.

[20] N. S. Arunraj, R. Hable, M. Fernandes, K. Leidl, and M. Heigl, "Comparison of supervised, semi-supervised and unsupervised learning methods in network intrusion detection system (nids) application," *Anwendungen und Konzepte der Wirtschaftsinformatik*, vol. 6, 2017.

[21] C. Alcaraz, "Secure interconnection of it-ot networks in industry 4.0," in *Critical Infrastructure Security and Resilience*. Springer, 2019, pp. 201–217.

[22] G. Culot, F. Fattori, M. Podrecca, and M. Sartor, "Addressing industry 4.0 cybersecurity challenges," *IEEE Engineering Management Review*, vol. 47, no. 3, pp. 79–86, 2019.

[23] M. N. Al-Mhiqani, R. Ahmad, W. Yassin, A. Hassan, Z. Z. Abidin, N. S. Ali, and K. H. Abdulkareem, "Cyber-security incidents: a review cases in cyber-physical systems," *Int. J. Adv. Comput. Sci. Appl*, no. 1, pp. 499–508, 2018.

[24] M. Chiarvesio and R. Romanello, "Industry 4.0 technologies and internationalization: Insights from italian companies," in *International Business in the Information and Digital Age*. Emerald Publishing Limited, 2018.

[25] J. P. Vilko and J. M. Hallikas, "Risk assessment in multimodal supply chains," *International Journal of Production Economics*, vol. 140, no. 2, pp. 586–595, 2012.

[26] Z. Rajnai and I. Kocsis, "Assessing industry 4.0 readiness of enterprises," in *2018 IEEE 16th world symposium on applied machine intelligence and informatics (SAMI)*. IEEE, 2018, pp. 000 225–000 230.

[27] M. D. Cavelty and F. J. Egloff, "The politics of cybersecurity: Balancing different roles of the state," *St Antony's International Review*, vol. 15, no. 1, pp. 37–57, 2019.

[28] B. Farrand and H. Carrapico, "Blurring public and private: cybersecurity in the age of regulatory capitalism," in *Security Privatization*. Springer, 2018, pp. 197–217.

[29] H. N. Alshabib and J. T. Martins, "Cybersecurity: Perceived threats and policy responses in the gulf cooperation council," *IEEE Transactions on Engineering Management*, 2021.

[30] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain for 5g-enabled iot for industrial automation: A systematic review, solutions, and challenges," *Mechanical Systems and Signal Processing*, vol. 135, p. 106382, 2020.

[31] A. Ahmadi, M. Moradi, C. Cherifi, V. Cheutet, and Y. Ouzrout, "Wireless connectivity of cps for smart manufacturing: A survey," in *2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*. IEEE, 2018, pp. 1–8.

[32] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[33] S. Petrenko, K. Makoveichuk, and A. Olifirov, "New methods of the cybersecurity knowledge management analytics," in *International Conference on Convergent Cognitive Information Technologies*. Springer, 2018, pp. 296–310.

[34] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial iot: Blockchain system with credit-based consensus mechanism," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3680–3689, 2019.

[35] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in *2012 IEEE symposium on security and privacy*. IEEE, 2012, pp. 95–109.

[36] M. Proyect. Android malware genome project. [Online]. Available: http://www.malgenomeproject.org

# CAPÍTULO 2

# Métodos

A continuación, se describirán los algoritmos empleados en este trabajo de investigación de forma detallada, así como las principales características y paramétros de ajuste necesarios. Como el objetivo de este trabajo es el empleo de las técnicas y no su análisis exhaustivo, no se definirán la totalidad de sus ajustes ni tampoco su funcionamiento en esos casos.

Los algoritmos empleados en este trabajo se enmarcan dentro de técnicas inteligentes de aprendizaje no supervisado y, en concreto, en redes neuronales para reducción dimensional y clustering (figura 2.1).

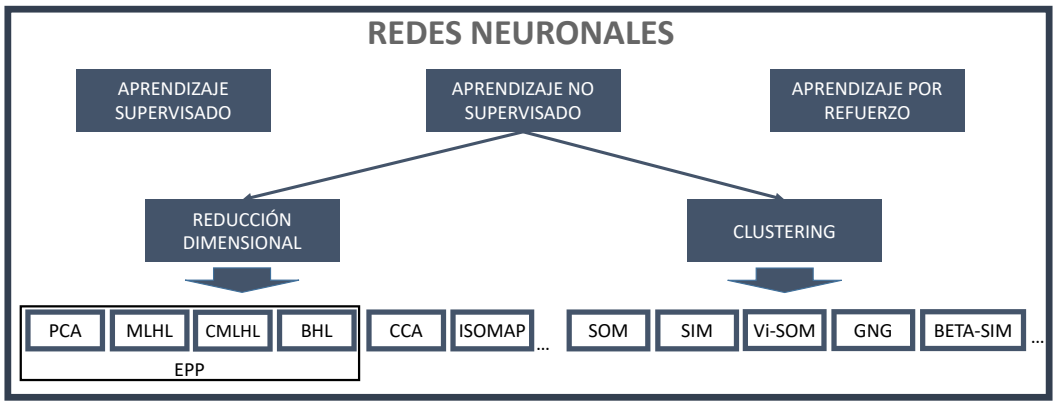


Figura 2.1: Clasificación de algoritmos de redes neuronales basado en el tipo de aprendizaje.

El principal objetivo de los algoritmos que se describen a continuación es el de dar solución al problema de identificar patrones que existen a través de los límites dimensionales en conjuntos de datos de alta dimensionalidad, siendo posible resolver dicho problema mediante un cambio en las coordenadas espaciales de los datos. Los métodos de proyección mapean puntos de datos de alta dimensión en un espacio de menor dimensión para identificar direcciones "interesantes" en términos de cualquier

índice o proyección específicos. Una vez identificadas las proyecciones más interesantes, los datos se proyectan luego en un subespacio de menor dimensión graficado en dos o tres dimensiones, lo que permite examinar la estructura a simple vista [1].

## 2.1. Principal Component Analysis

El Análisis de Componentes Principales (PCA) es un modelo estadístico [2] que describe la variación en un conjunto de datos multivariados en términos de un conjunto de variables no correlacionadas, cada una de las cuales es una combinación lineal de variables originales. Desde un punto de vista geométrico, su objetivo consiste principalmente en una rotación de los ejes del sistema de coordenadas original a un nuevo conjunto de ejes ortogonales que están ordenados en términos de la cantidad de varianza de los datos originales que representan.

PCA puede ser implementado mediante modelos neuronales como los descritos en [3] o [4]. Cabe señalar que incluso si se consigue caracterizar los datos con unas pocas variables (componentes principales), estas nuevas variables obtenidas con PCA no tienen porqué tener una interpretación. En la figura 2.2 puede verse una representación gráfica de las 2 primeras componentes del espacio de salida tras aplicar PCA.

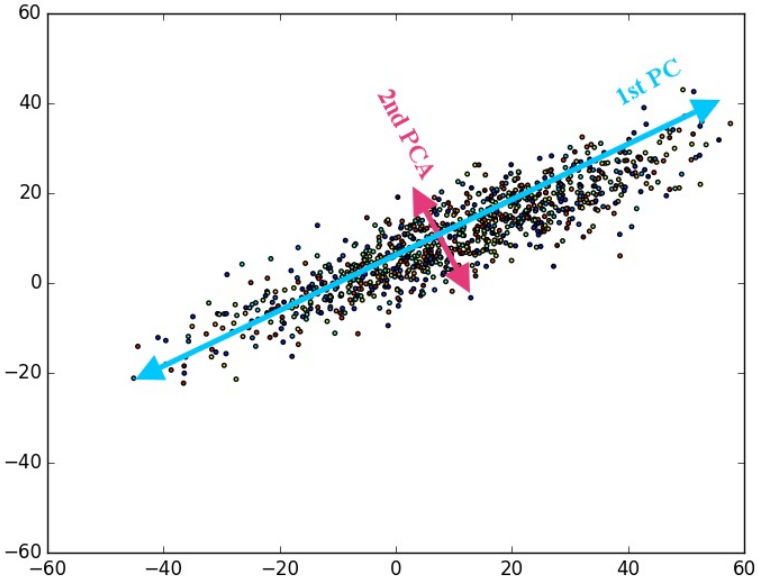


Figura 2.2: Representación gráfica de las 2 primeras componentes de PCA sobre un conjunto de datos.

Desde un punto de vista matemático se puede describir como mapear vectores $x^d$ en un espacio N-dimensional $(x_1, ..., x_n)$ sobre vectores $y^d$ en un espacio M-dimensional $(y_1, ..., y_m)$, donde $M \leq N$. $x$ puede representarse como una combinación lineal de un conjunto de N vectores ortonormales $W_i$ (ecuación 2.1).

$$x = \sum_{i=1}^{N} y_i \cdot W_i \tag{2.1}$$

Los vectores $W_i$ satisfacen la relación de ortonormalidad presentada en la ecuación 2.2.

$$W_i^t \cdot W_j = \delta_{i,j} \tag{2.2}$$

donde $\delta_{i,j}$ es el delta de Kronecker.

Haciendo uso de la expresión ecuación 2.1, los coeficientes $y_i$ pueden estar dados por la ecuación 2.3.

$$y_i = W_i^t \cdot x \tag{2.3}$$

que puede considerarse como una simple rotación del sistema de coordenadas de las $x$ originales a un nuevo conjunto de coordenadas dadas por las $y$. Donde $W_i$ son los vectores propios de la matriz de covarianza del conjunto de datos sobre los que se aplica PCA. Se puede demostrar que los vectores propios son ortogonales. Por lo tanto, el error mínimo se obtiene eligiendo los $(N - M)$ valores propios más pequeños, y sus correspondientes vectores propios. Por lo general, se nombran las $y$ como los componentes principales.

## 2.2. Maximum Likelihood Hebbian Learning

Maximum Likelihood Hebbian Learning (MLHL) [1] es un modelo de EPP cuya familia de reglas de aprendizaje se basa en distribuciones exponenciales. El método estadístico de EPP fue diseñado para resolver el complejo problema de identificar la

estructura en datos de alta dimensión proyectándola en un subespacio de menor dimensión en el que se busca su estructura de forma visual. Para ello, se debe definir un "índice" para medir los distintos grados de interés asociados con cada proyección. Posteriormente, los datos se transforman maximizando el índice y el interés asociado. Desde un punto de vista estadístico, las direcciones más interesantes son aquellas que son lo más no gaussianas posible.

Para ello se define una función de coste asociada MLHL tal como se muestra en la ecuación

$$J = \mathbb{E}(-log(p(e))) = \mathbb{E}(|e|^p + K) \tag{2.4}$$

donde K es una constante independiente de $W$ (pesos de la red MLHL) y la expectativa se toma sobre el conjunto de datos de entrada, y $\mathbb{E}$ es el operador de valor esperado. Por lo tanto, el descenso del gradiente $J$ se presenta en la equación 2.5.

$$\Delta W - \frac{\partial J}{\partial W}\bigg|_{W_{t-1}} = -\frac{\partial J}{\partial e}\frac{\partial e}{\partial W}\bigg|_{W_{(t-1)}} \approx \mathbb{E}\{\, y(p\,|e|^{p-1}\, sign(e))^T \big|_{W_{(t-1)}}\} \tag{2.5}$$

$W_{t-1}$ son los pesos en el instante de tiempo $t-1$ y $T$ denota la transposición de un vector.

Si se satisfacen las condiciones de aproximación estocástica, la media puede aproximarse con una ecuación en diferencias, por lo que la regla de actualización de pesos puede aproximarse a la ecuación 2.6

$$\Delta W_{ij} = \mu \cdot y_i \cdot sign(e_j) \cdot |e_j|^{p-1} \tag{2.6}$$

donde $\Delta W_{ij}$ es la actualización de pesos de la red, $e$ el residuo, $\mu$ es la tasa de aprendizaje e $y$ es la salida de la red. Se espera que para los residuos leptocúrticos (más kurtóticos que una distribución gaussiana), los valores de $p < 2$ serían apropiados, mientras que para los residuos platicúrticos (menos kurtóticos que un gaussiano), los valores de $p > 2$ serían apropiados. Finalmente, el funcionamiento de la red se puede expresar mediante las ecuaciones 2.7, 2.8 y 2.9.

$$Feedforward : y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{2.7}$$

$$Feedback : e_j = x_j - \sum_{i=1}^{M} W_{ij} y_j, \forall j \tag{2.8}$$

$$Actualización\ de\ pesos : \Delta W_{ij} = \eta \cdot y_i \cdot sign(e_j)|e_j|^{p-1} \tag{2.9}$$

## 2.3. Cooperative Maximum Likelihood Hebbian Learning

Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [5] extiende el modelo MLHL, agregando conexiones laterales entre neuronas en la capa de salida del modelo. Considerando un vector de entrada $N - dimensional$ $(x)$ y un vector de salida $M - dimensional$ $(y)$, con $W_{ij}$ como el peso que une la neurona de entrada $i$ con la neurona de salida $j$, entonces CMLHL se puede expresar mediante las ecuaciones 2.10, 2.11, 2.12 y 2.13.

1. Feed-forward:

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{2.10}$$

2. Activación de conexiones laterales:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \tag{2.11}$$

3. Feedback:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_j, \forall j \tag{2.12}$$

4. Actualización de pesos

$$\Delta W_{ij} = \eta \cdot y_i \cdot sign(e_j)|e_j|^{p-1} \tag{2.13}$$

donde $\eta$ es el coeficiente de aprendizaje, $\tau$ la "fuerza" de las conexiones laterales, $b$ el parámetro *bias*, $p$ un parámetro relacionado con la función de coste de energía y $A$ la matriz asimétrica usada para modificar la respuesta a los datos. El efecto de esta matriz se basa en la relación entre las distancias que separan las neuronas de salida.

## 2.4. ISOMAP

El algoritmo no lineal ISOMAP DRT [6] intenta preservar la distancia geodésica (o curvilínea) por pares entre puntos de datos. La distancia geodésica es la distancia entre dos puntos medidos a través del camino más corto. ISOMAP define la distancia geodésica como la suma de los pesos de los bordes a lo largo del camino más corto entre dos nodos (ver figura 2.3).

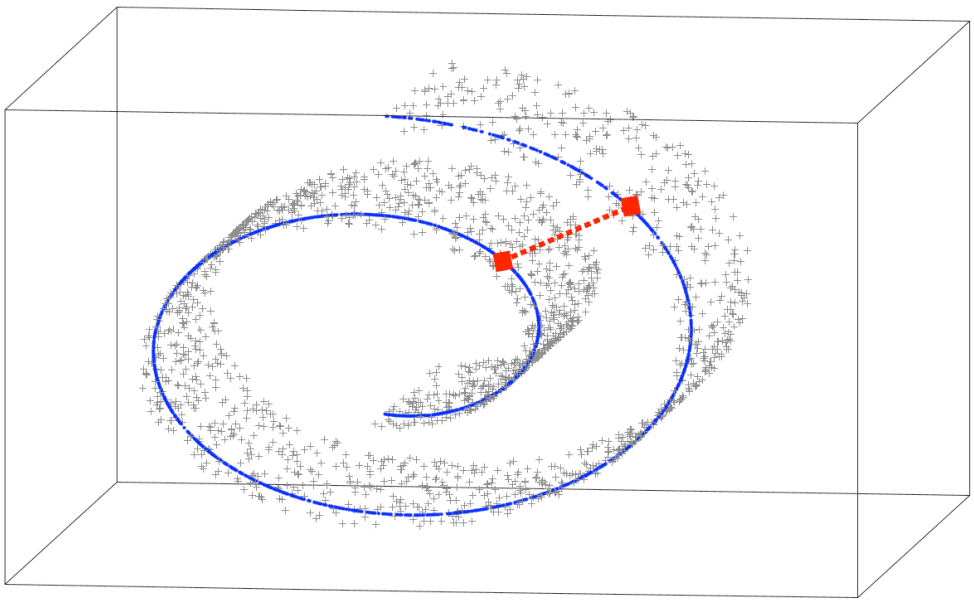

Figura 2.3: Descripción de la distancia geodésica sobre el conjunto de datos Swiss roll.

La matriz de distancias geodésicas doblemente centrada $K$ en ISOMAP tiene la forma dada por la ecuación 2.14.

$$K = \frac{1}{2} H D^2 H \tag{2.14}$$

donde $D^2 = D_{ij}^2$ es el elemento al cuadrado de la matriz de distancia geodésica $D = [D_{ij}]$, y $H$ es la matriz de centrado, dada por la ecuación 2.15.

$$H = I_n - \frac{1}{N} e_N e_N^T \tag{2.15}$$

donde $e_N = [1...1]^T \; en \; R^N$

Los vectores superiores $N$ de la matriz de distancia geodésica representan las coordenadas en el nuevo espacio euclidiano $n-$ dimensional.

## 2.5. Curvilinear Component Analysis Algorithm

*Curvilinear Component Analisys* (CCA) [7], [8] es un método de proyección no lineal que conserva las relaciones de distancia en los espacios de entrada y salida. CCA es un método útil para la representación de estructuras de datos redundantes y no lineales y se puede utilizar en la reducción de dimensionalidad. CCA es útil con datos altamente no lineales, donde PCA o cualquier otro método lineal no proporciona información adecuada.

CCA mejora a otros métodos como Sammon's Mapping [9], aunque cuando se despliega una estructura no lineal, el Sammon's Mapping no puede reproducir todas las distancias. Una forma de sortear este problema consiste en favorecer la topología local: CCA intenta reproducir distancias cortas en primer lugar, siendo las distancias largas secundarias. Formalmente, este razonamiento condujo a la siguiente función de error (sin normalización) definida en la ecuación 2.16.

$$E_{CCA} = \sum_{i,j=1}^{N} (d_{i,j}^n - d_{i,j}^p)^2 F_\lambda(d_{i,j}^p) \tag{2.16}$$

En comparación con $E_{Sammon}$, $E_{CCA}$ tiene una función de ponderación adicional $F$ dependiendo de $d_{ij}^p$ y del parámetro $\lambda$. El factor $F$ es un factor decreciente función de su argumento, por lo que se utiliza para favorecer la preservación de la topología local. Por ejemplo, $F$ podría ser una función escalonada de ($\lambda$-d).

## 2.6. Self Organizing Maps

Entre la gran variedad de herramientas para la visualización de datos multidimensionales, varias de las más utilizadas son las que pertenecen a la familia de mapas de preservación de topología [10] - [11]. Probablemente el más conocido entre estos algoritmos es el mapa autoorganizado (SOM) [10], [12], [13], [14], el cual se basa en un tipo de aprendizaje no supervisado llamado aprendizaje competitivo; proceso adaptativo en el que las unidades de una red neuronal se vuelven gradualmente sensibles a diferentes categorías de entrada o conjuntos de muestras en un dominio específico del espacio de entrada. La característica principal del algoritmo SOM es la preservación de su topología. Cuando no sólo se permite aprender a la unidad ganadora, sino también a sus vecinos en la red, las unidades vecinas se especializan gradualmente para representar entradas similares y las representaciones se ordenan en la red del mapa.

Cada unidad $j$ tiene un vector característico $d$-dimensional asociado $w_j = [w_{j1}, ..., w_{jd}]$. Las posiciones de la unidad $k_j$ en la cuadrícula están fijas desde el principio. El mapa se ajusta a los datos adaptando los vectores prototipo. Juntos, la cuadrícula y el conjunto de vectores de características forman un mapa de baja dimensión: una representación bidimensional donde los objetos topológicamente relacionados (unidades de mapa o neuronas) están cerca unos de otros (ver figura 2.4).

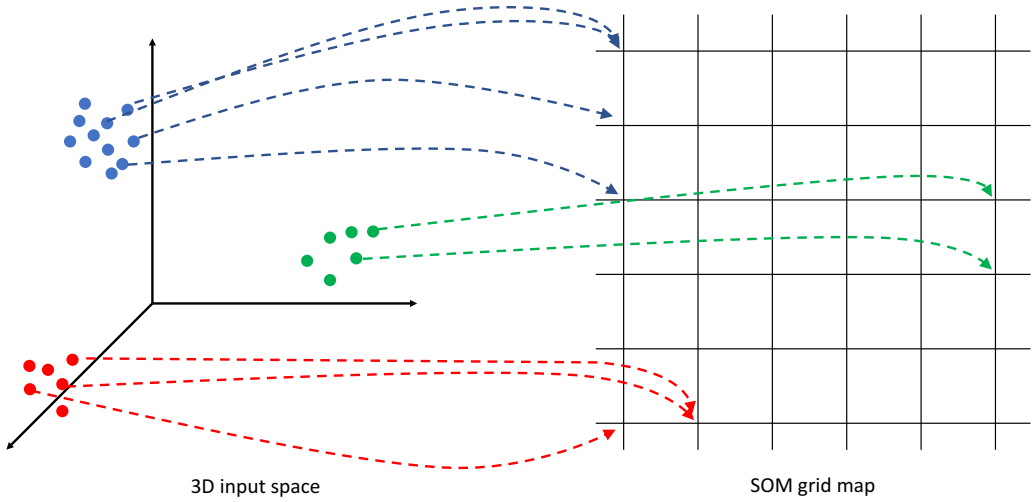


Figura 2.4: Mapeado del conjunto de datos de entrada (3D) sobre una red SOM

Se presenta un vector de entrada ($x$) a la red y se elige el nodo de la red en el que los pesos ($W_i$) son los más cercanos (en términos de distancia euclidiana) a $x$:

$$c = argmin(\| x - W_i \|) \tag{2.17}$$

Los pesos del nodo ganador (BMU) y los nodos cercanos a él se actualizan para acercarse al vector de entrada (ver figura 2.5).

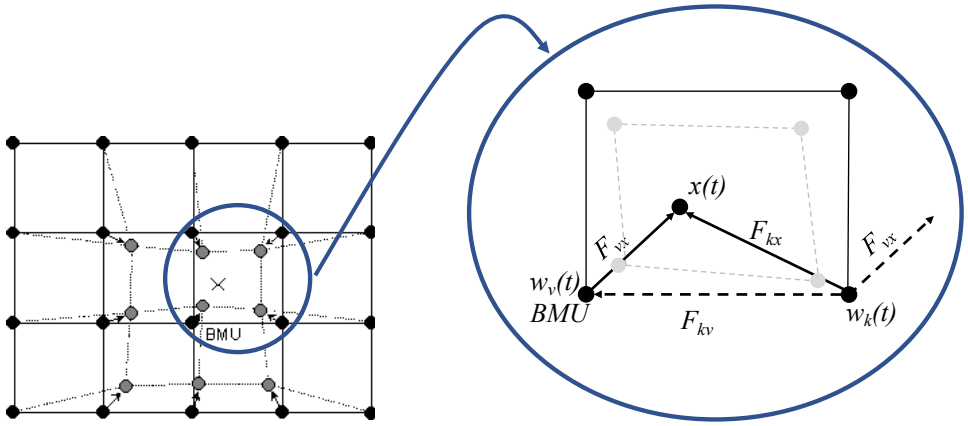


Figura 2.5: Actualización de pesos de las neuronas en función de la BMU.

También hay un parámetro de tasa de aprendizaje que generalmente disminuye a medida que avanza el proceso de entrenamiento. La regla de actualización de peso para las entradas se define de la siguiente manera:

$$\Delta W_i = \eta h_{ci}(x - W_i), \ \forall i \in N \tag{2.18}$$

Donde, $W_i$ es el vector de peso asociado con la neurona $i$, $x$ es el vector de entrada y $h$ es la función de vecindad.

## 2.7. Beta Hebbian Learning

El algoritmo Beta Hebbian Learning (BHL) [15] es una red neuronal no supervisada de la familia de EPP que emplea la distribución Beta para actualizar su regla de aprendizaje y ajustar la función de densidad de probabilidad (PDF) del residual con la distribución de un conjunto de datos dado.

Por lo tanto, si se conoce la PDF de los residuos, se puede determinar la función de costo óptima. Al usar los parámetros $B(\alpha, \beta)$ de la distribución Beta, el residuo $(e)$ se puede obtener con la siguiente PDF (ver ecuación 2.19):

$$p(e) = e^{\alpha-1}(1-e)^{\beta-1} = (x - Wy)^{\alpha-1}(1 - x + Wy)^{\beta-1} \tag{2.19}$$

donde $\alpha$ y $\beta$ se usan para ajustar la forma del PDF de la distribución Beta, $x$ es la entrada de la red, $e$ es el residuo, $W$ es la matriz de pesos e $y$ es la salida de la red.

Luego, usando la ecuación 2.20, el descenso de gradiente se realiza para maximizar la probabilidad de los pesos:

$$\begin{aligned}\frac{\partial p}{\partial W} = (e_j^{\alpha-2}(1-e_j)^{\beta-2}(-(\alpha-1)(1-e_j) + e_j(\beta-1))) = \\ (e_j^{\alpha-2}(1-e_j)^{\beta-2}(1-\alpha + e_j(\alpha+\beta-2)))\end{aligned} \tag{2.20}$$

En el caso de BHL, la regla de aprendizaje permite ajustar la PDF del residuo, maximizando la probabilidad de dicho residuo con la distribución actual.
Finalmente la arquitectura neuronal para BHL se define de la siguiente forma:

$$Feedforward : y_i = \sum_{j=1}^{N} W_{ij}x_j, \, \forall i \tag{2.21}$$

$$Feedback : e_j = x_j - \sum_{i=1}^{M} W_{ij}y_i \tag{2.22}$$

$$Actualización\ de\ pesos : \Delta W_{ij} = \eta(e_j^{\alpha-2}(1-e_j)^{\beta-2}(1-\alpha + e_j(\alpha+\beta-2)))y_i \tag{2.23}$$

## Bibliografía

[1] E. Corchado, D. MacDonald, and C. Fyfe, "Maximum and minimum likelihood hebbian learning for exploratory projection pursuit," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 203–225, 2004.

[2] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[3] E. Oja, "Principal components, minor components, and linear neural networks," *Neural networks*, vol. 5, no. 6, pp. 927–935, 1992.

[4] C. Fyfe, "A neural network for pca and beyond," *Neural Processing Letters*, vol. 6, no. 1-2, pp. 33–41, 1997.

[5] E. Corchado and C. Fyfe, "Connectionist techniques for the identification and suppression of interfering underlying factors," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 08, pp. 1447–1466, 2003.

[6] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. I–I.

[7] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on neural networks*, vol. 8, no. 1, pp. 148–154, 1997.

[8] G. Cirrincione, J. Hérault, and V. Randazzo, "The on-line curvilinear component analysis (oncca) for real-time data reduction," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[9] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

[10] N. Chen, B. Ribeiro, A. Vieira, and A. Chen, "Clustering and visualization of bankruptcy trajectory using self-organizing map," *Expert Systems with Applications*, vol. 40, no. 1, pp. 385–393, 2013.

[11] H. Quintián and E. Corchado, "Beta scale invariant map," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 218–235, 2017.

[12] T. Kohonen, "The self-organizing map. neurocomputing 21, 1e6," 1998.

[13] H. FAKHOURI, L. CHERRAT, and M. Ezziyyani, "Towards a new approach to improve the classification accuracy of the kohonen's self-organizing map during learning process."

[14] T. Kohonen, "Essentials of the self-organizing map," *Neural networks*, vol. 37, pp. 52–65, 2013.

[15] H. Quintián and E. Corchado, "Beta hebbian learning as a new method for exploratory projection pursuit," *International Journal of Neural Systems*, vol. 27, no. 6, pp. 1–16, 2017. [Online]. Available: https://doi.org/10.1142/S0129065717500241

# CAPÍTULO 3

# Descripción de la propuesta

Este capítulo describe de forma detallada la propuesta presentada en este trabajo, explicando el proceso aplicado para poder obtener un sistema que finalmente permita los objectivos anteriormente expuestos.

En el presente trabajo se emplearán técnicas de aprendizaje no supervisado bajo la perspectiva de la visualización de la estructura interna de los datos, que a diferencia de los métodos de aprendizaje supervisado, intenta representar todos los datos de una manera intuitiva de modo que aquellos datos anómalos puedan ser identificados a simple vista. Esto se basa en la capacidad innata del ser humano de identificar visualmente patrones anómalos.

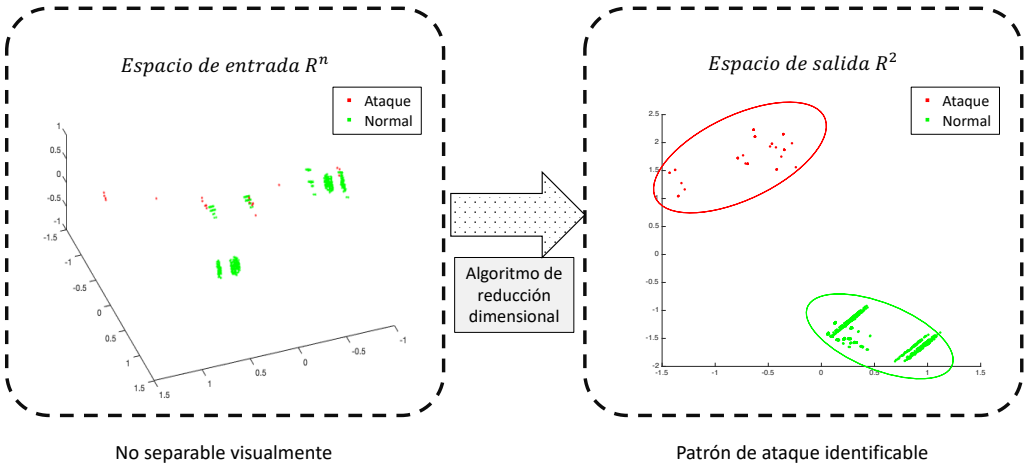


Figura 3.1: Reducción dimensional.

Es por ello que el sistema propuesto en este trabajo consistirá en analizar de forma visual los conjuntos de datos del presente trabajo, en una menor dimensionalidad de la original buscando visualmente patrones que representen ataques en redes informáticas (figura 3.1).

En un primer paso se realizará un preprocesado de los datos que consistirá en la eliminación de aquellos que sean erróneos o incompletos para una posterior normalización de los mismos. A continuación, se aplicarán diversos algoritmos de reducción dimensional para obtener proyecciones en nuevos subespacios de menor dimensionalidad de modo que los patrones internos de los conjuntos de datos de alta dimensionalidad ($R^n$), sean visibles en una menor dimensión ($R^2$), de esta forma se conseguirá un sistema que permitirá determinar visualmente cuando un nuevo dato es constituyente de ser un ataque al sistema, ya que este se proyectará en áreas que no corresponden al funcionamiento normal de la red informática (figura 3.2).

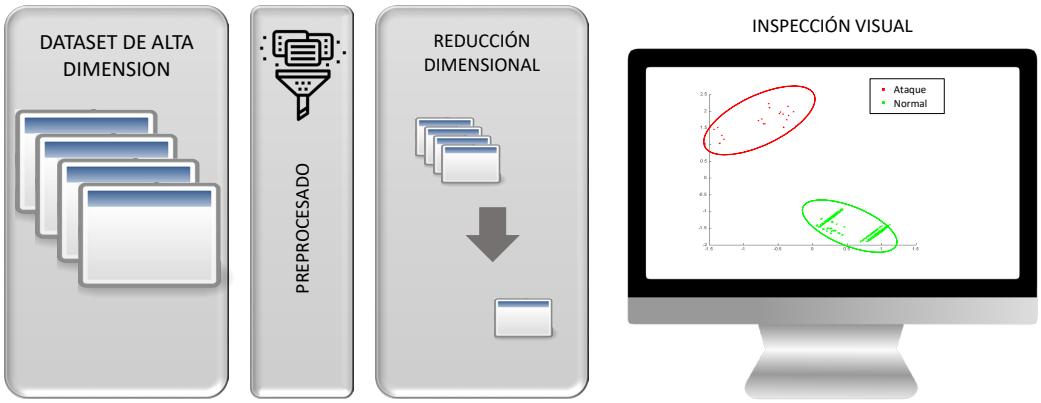


Figura 3.2: Esquema del sistema propuesto.

# CAPÍTULO 4

# Gaining Deep Knowledge of Android Malware Families through

En este capítulo se presenta el primero de los artículos utilizado para la defensa de la tesis por compendio de artículos. A continuación se presentan los datos básicos del mismo, y después de incluirá todo el contenido del artículo copiado del artículo publicado. Se ha decidido no incluir el artículo original para mantener una uniformidad en la presentación de la tesis, pero se puede consultar a través del *doi* correspondiente.

Datos generales del artículo:

**Autores:**
Rafael Vega Vega[1], Héctor Quintián[1], José Luis Calvo-Rolle[1], Álvaro Herrero[2], and Emilio Corchado[3]
**Afiliaciones:**
[1] Department of Industrial Engineering, University of A Coruña, A Coruña, Spain
Avda. 19 de febrero S/N 15.405, Ferrol - Coruña, Spain
ravega@udc.es, hector.quintian@udc.es, jlcalvo@udc.es
[2] Department of Civil Engineering, University of Burgos, Spain
Avenida de Cantabria s/n, 09006 Burgos, Spain
ahcosio@ubu.es
[3] Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced, s/n, 37008 Salamanca, Spain
escorchado@usal.es
**Título:** Gaining Deep Knowledge of Android Malware Families through Dimensionality Reduction Techniques
**Revista:** LOGIC JOURNAL OF THE IGPL
**Publicación:** Vol. 27, no. 2
**Páginas:** 160–176
**Editorial:** Oxford University Press
**Año de publicación:** 2019
**DOI:** 10.1093/jigpal/jzy030
**Factor de impacto JCR 2019:** 0,931
**Factor de impacto JCR 2019 (5 años):** 0,706
**Disciplina *Logic (SCIE)* 2019:** 3/21 - Q1
**Disciplina *Mathematics (SCIE)* 2019:** 126/325 - Q2
**Disciplina *Mathematics, Applied (SCIE)* 2019:** 171/261 - Q3

## Abstract

This research proposes the analysis and subsequent characterization of Android malware families, by means of low dimensional visualizations using dimensional reduction techniques. The well-known Malgenome dataset, coming from the Android Malware Genome Project, has been thoroughly analysed through six dimensionality reduction techniques: Principal Component Analysis, Maximum Likelihood Hebbian Learning, Cooperative Maximum Likelihood Hebbian Learning, Curvilinear Component Analysis, Isomap and Self Organizing Map. Results obtained enable a clear visual analysis of the structure of this high-dimensionality dataset, letting us gain deep knowledge about the nature of such Android malware families. Interesting conclusions are obtained from the real-life dataset under analysis.

**Keywords:** Android Malware, Malware Families, Dimensionality Reduction, Artificial Neural Networks

## 4.1. Introduction

Since the first smartphones came onto the market in the late 1990s, sales on that sector have increased constantly to the present-day. Among all the available operating systems, Google's Android has been, and increasingly is, the most popular mobile platform [1]. The number of Android units sold in Q1 2017 worldwide raised to 379.98 million out of 432.79 million units, that is a share of 87.79 %. It is not only the number of devices but also the number of apps; those available at Google Play (Android's official store) constantly increase, up to more than 3.4 million that are available nowadays [2]. With regard to the security issue, the number of malicious Android apps has greatly risen in the last four years; from the half million of them that were identified in 2013 to the nearly 3.5 million in 2017 [3] . Furthermore, it has been forecast that increase in malware for Android devices is expected to continue [3, 4]. This operating system is an appealing target for bad-intentioned apps, mainly because of its open mentality, in contrast to iOS or some other operating systems.

Smartphone security and privacy still are nowadays major concerns although great efforts have been devoted over past years [5]. In order to address these issues, it is required to understand the malware and its nature. Otherwise, it will not be possible

to practically develop an effective solution [6]. According to this idea of gaining deeper knowledge about malware nature, present study is focused on the analysis of Android malware families. To do so, Malgenome (a real-life publicly-available) dataset [7] has been analyzed by means of several Dimensionality Reduction Techniques (DRTs). From the samples contained in such dataset, several alarming statistics were found [6], that motivate further research on Android malware. That is the case of the 36.7 % of the collected samples that leverage root-level exploits to fully compromise the security of the whole system or the fact that more than 90 % of the samples turn the compromised phones into a botnet controlled through network or short messages.

To characterize malware families, present study proposes a comprehensive comparison of many DRTs, that are able to visualize a high-dimensionality dataset (further described in section 2), to gain deep knowledge of Android malware families. Each individual from the Malgenome dataset (a malware app) encodes the subset of selected features by using a binary representation (details on section 3). These individuals are grouped by families and then visualized trying to identify patterns that exist across dimensional boundaries in the high dimensional dataset by changing the spatial coordinates of malware family data. The main goal is to obtain an intuitive visualization of the malware families to draw conclusions about the structure of the dataset and to characterize malware families subsequently.

Neural networks have been applied to a wide variety of fields in recent decades [8],[9],[10],[11],[12]; additionally, neural DRTs have been previously applied to massive security datasets, such as those generated by network traffic [13], [14], SQL code [15], [16], honeynets [17], and HTTP traffic [18]. In present paper, such methods are applied to a new problem, related to the characterization and knowing of malware families. On the other hand, several different techniques have been used to differentiate between legitimate and malicious Android apps, such as machine learning [19],[20],[21], knowledge discovery [22], and weighted similarity matching of logs [23], among others as well as hybridization approaches [24]. Although some visualization techniques have been applied to the detection of malware in general terms [25], few dimensionality-reduction proposals for Android malware detection are available at present time. In [26] Pythagoras tree fractal is used to visualize the malware data, being all apps scattered, as leaves in the tree. Authors of [27] proposed graphs for deciding about malware by depicting lists of malicious methods, needless permissions and malicious strings. In [28], visualization obtained from biclustering on permission information is described. Behavior-related dendrograms are generated out of malware traces in [29], comprising nodes related to the package name of the application, the Android components that

has called the API call and the names of functions and methods invoked by the application. Unlike previous work, Android malware families (instead of malware apps) are visualized by DRTs in present paper. Up to the authors knowledge, this is the first time that dimensionality-reduction models are applied to visualize Android malware.

The rest of this paper is organized as follows: the applied neural methods are described in section 2, the setup of experiments for the Android Malware Genome dataset is described in section 3, together with the results obtained and the conclusions of the study that are stated in section 4.

## 4.2. Dimensionality Reduction Techniques

This work proposes the application of several DRTs for the visualization of Android malware data. Visualization techniques are considered a viable approach to information seeking, as humans are able to recognize different features and to detect anomalies by means of visual inspection [30]. The underlying operational assumption of the proposed approach is mainly grounded in the ability to render the high-dimensional traffic data in a consistent yet low-dimensional representation [17], [18], [25]. In most cases, security visualization tools have to deal with massive datasets with a high dimensionality, to obtain a low-dimensional space for presentation [13], [15], [17], [18], [31], [32].

This problem of identifying patterns that exist across dimensional boundaries in high dimensional datasets can be solved by changing the spatial coordinates of data. Projection methods project high-dimensional data points onto a lower dimensional space in order to identify ïnteresting"directions in terms of any specific index or projection. Having identified the most interesting projections, the data are then projected onto a lower dimensional subspace plotted in two or three dimensions, which makes it possible to examine the structure with the naked eye [30].

### 4.2.1. Principal Component Analysis

Principal Component Analysis (PCA) is a well-known statistical model, introduced in [33], that describes the variation in a set of multivariate data in terms of a set of uncorrelated variables each, of which is a linear combination of the original variables.

From a geometrical point of view, this goal mainly consists of a rotation of the axes of the original coordinate system to a new set of orthogonal axes that are ordered in terms of the amount of variance of the original data they account for.

PCA can be performed by means of neural models such as those described in [34] or [35]. It should be noted that even if we are able to characterize the data with a few variables, it does not follow that an interpretation will ensue.

### 4.2.2. Maximum Likelihood Hebbian Learningl

Maximum Likelihood Hebbian Learning [30] which is based on Exploration Projection Pursuit (EPP). The statistical method of EPP [30], [36], [37] was designed for solving the complex problem of identifying structure in high dimensional data by projecting it onto a lower dimensional subspace in which its structure is searched for by eye. To that end, an "index" must be defined to measure the varying degrees of interest associated with each projection. Subsequently, the data is transformed by maximizing the index and the associated interest. From a statistical point of view the most interesting directions are those that are as non-Gaussian as possible.

### 4.2.3. Cooperative Maximum Likelihood Hebbian Learning

The Cooperative MLHL (CMLHL) model [38] extends the MLHL model, by adding lateral connections between neurons in the output layer of the model. Considering an N-dimensional input vector (x), and an M-dimensional output vector (y), with Wij being the weight (linking input neuron j to output neuron i), then CMLHL can be expressed as defined in equations 1-4.

1. Feed-forward step:

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{4.1}$$

2. Lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \tag{4.2}$$

3. Feedback step:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_j, \forall j \tag{4.3}$$

4. Weight change

$$\Delta W_{ij} = \eta \cdot y_i \cdot sign(e_j)|e_j|^{p-1} \tag{4.4}$$

Where: $\eta$ is the learning rate, $\tau$ is the "strength" of the lateral connections, $b$ the bias parameter, $p$ a parameter related to the energy function and $A$ a symmetric matrix used to modify the response to the data. The effect of this matrix is based on the relation between the distances separating the output neurons.

### 4.2.4. ISOMAP Algorithm

ISOMAP nonlinear DRT [39] attempts to preserve pairwise geodesic (or curvilinear) distance between data points. Geodesic distance is the distance between two points measured over the manifold. ISOMAP defines the geodesic distance as the sum of edge weights along the shortest path between two nodes. The doubly-centered geodesic distance matrix $K$ in ISOMAP is of the form given by equation 4.5.

$$K = \frac{1}{2} H D^2 H \tag{4.5}$$

Where $D^2 = D_{ij}^2$ means the element wise square of the geodesic distance matrix $D = [Dij]$, and H is the centring matrix, given by equation 4.6.

$$H = I_n - \frac{1}{N} e_N e_N^T \tag{4.6}$$

In which $e_N = [1...1]^T \in R^N$

The top $N$ eigenvectors of the geodesic distance matrix represent the coordinates in the new $n-$dimensional Euclidean space.

### 4.2.5. Curvilinear Component Analysis Algorithm

Curvilinear Component Analysis (CCA) [40], [41] is a non-linear projection method that preserves distance relationships in both input and output spaces. CCA is a useful

method for redundant and non-linear data structure representation and can be used in dimensionality reduction. CCA is useful with highly non-linear data, where PCA or any other linear method fails to give suitable information.

CCA brings some improvements to other methods like Sammon's Mapping [42], although when unfolding a nonlinear structure, Sammon's Mapping cannot reproduce all distances. One way to get round this problem consists in favoring local topology: CCA tries to reproduce short distances firstly, long distances being secondary. Formally, this reasoning led to the following error function (without normalization) defined in equation 4.7.

$$E_{CCA} = \sum_{i,j=1}^{N} (d_{i,j}^{n} - d_{i,j}^{p})^2 F_{\lambda}(d_{i,j}^{p}) \tag{4.7}$$

In comparison with $E_{Sammon}$, $E_{CCA}$ has an additional weighting function $F$ depending on $d_{ij}^{p}$ and on parameter $\lambda$ .The $F$ factor is a decreasing function of its argument, so it is used to favour local topology preservation. For example, $F$ could be a step function of ($\lambda$ -d).

### 4.2.6. Self Organizing Maps

Among the great variety of tools for multidimensional data visualization, several of the most widely used are those belonging to the family of the topology preserving maps [43],[44],[45],[46],[47],[48]. Probably the best known among these algorithms is the Self-Organizing Map (SOM) [43], [45], [49], [50]. It is based on a type of unsupervised learning called competitive learning; an adaptive process in which the units in a neural network gradually become sensitive to different input categories or sets of samples in a specific domain of the input space. The main feature of the SOM algorithm is its topology preservation. When not only the winning unit, but also its neighbors on the lattice are allowed to learn, neighboring units gradually specialize to represent similar inputs, and the representations become ordered on the map lattice.

An input vector ($x$) is presented to the network and the node of the network in which the weights ($W_i$) are closest (in terms of Euclidean distance) to $x$, is chosen:

$$c = argmin(\parallel x - W_i \parallel) \tag{4.8}$$

The weights of the winning node and the nodes close to it are then updated to move closer to the input vector. There is also a learning rate parameter that usually decreases as the training process progresses. The weight update rule for inputs is defined as follows:

$$\Delta W_i = \eta h_{ci}(x - W_i), \ \forall i \in N \tag{4.9}$$

Where,$W_i$ is the weight vector associated with neuron $i$, $x$ is the input vector, and $h$ is the neighborhood function.

## 4.3. Experiments & Results

As previously mentioned, several different DRTs (see Section 2) have been applied to analyze Android malware. Present section introduces the analyzed dataset as well as the main obtained results.

### 4.3.1. Malgenome Dataset

The Malgenome dataset [6], coming from the Android Malware Genome Project [7], has been analysed in present study. It is the first large collection of Android malware (1,260 samples) that was split in malware families (49 different ones). It covered the majority of existing Android malware, collected from the beginning of the project in August 2010.

Data related to many different apps from a variety of Android app repositories were accumulated over more than one year. Additionally, malware apps were thoroughly characterized based on their detailed behavior breakdown, including the installation, activation, and payloads.

Collected malware was split in families, that were obtained by "carefully examining the related security announcements, threat reports, and blog contents from existing mobile antivirus companies and active researchers as exhaustively as possible and diligently requesting malware samples from them or actively crawling from existing official and alternative Android Markets" [6]. The defined families are: *ADRD, AnserverBot, Asroot, BaseBridge, BeanBot, BgServ, CoinPirate, Crusewin, DogWars, DroidCoupon, DroidDeluxe, DroidDream, DroidDreamLight, DroidKungFu1, DroidKungFu2, DroidKung-*

*Fu3, DroidKungFu4, DroidKungFuSapp, DoidKungFuUpdate, Endofday, FakeNetflix, FakePlayer, GamblerSMS, Geinimi, GGTracker, GingerMaster, GoldDream, Gone60, GPSSMSSpy, HippoSMS, Jifake, jSMSHider, Kmin, Lovetrap, NickyBot, Nickyspy, Pjapps, Plankton, RogueLemon, RogueSPPush, SMSReplicator, SndApps, Spitmo, TapSnake, Walkinwat, YZHC, zHash, Zitmo, and Zsone.* Samples of 14 of the malware families were obtained from the official Android market, while samples of 44 of the families came from unofficial markets. As some families are present in both markets (official and unofficial), the final dataset to be analysed consists of 49 samples (one for each family) and each sample is described by 26 different features derived from a study of each one of the apps. The features are divided into six categories, as can be seen in Table 4.1.

Tabla 4.1: Features describing each one of the malware families in the Malgenome dataset.

| **Category #1: Installation** | **Category #3: Privilege escalation** |
|---|---|
| 1 Repackaging | 14 exploid |
| 2 Update | 15 RATC/zimperlich |
| 3 Drive-by download | 16 ginger break |
| 4 Standalone | 17 asroot |
| **Category #2: Activation** | 18 encrypted |
| 5 BOOT | **Category #4: Remote control** |
| 6 SMS | 19 NET |
| 7 NET | 20 SMS |
| 8 CALL | **Category #5: Financial charges** |
| 9 USB | 21 phone call |
| 10 PKG | 22 SMS |
| 11 BATT | 23 block SMS |
| 12 SYS | **Category #6: Personal information stealing** |
| 13 MAIN | 24 SMS |
| | 25 phone number |
| | 26 user account |

The features describing each family take the values of 0 (if that feature is not present in that family) or 1 (if the feature is present).

### 4.3.2. Results

For comparison purposes, some different projection models have been applied, whose results are shown below.

**PCA Projection**

Fig. 1 shows the principal component projection (components 1 and 2), obtained by applying PCA to the previously described data.

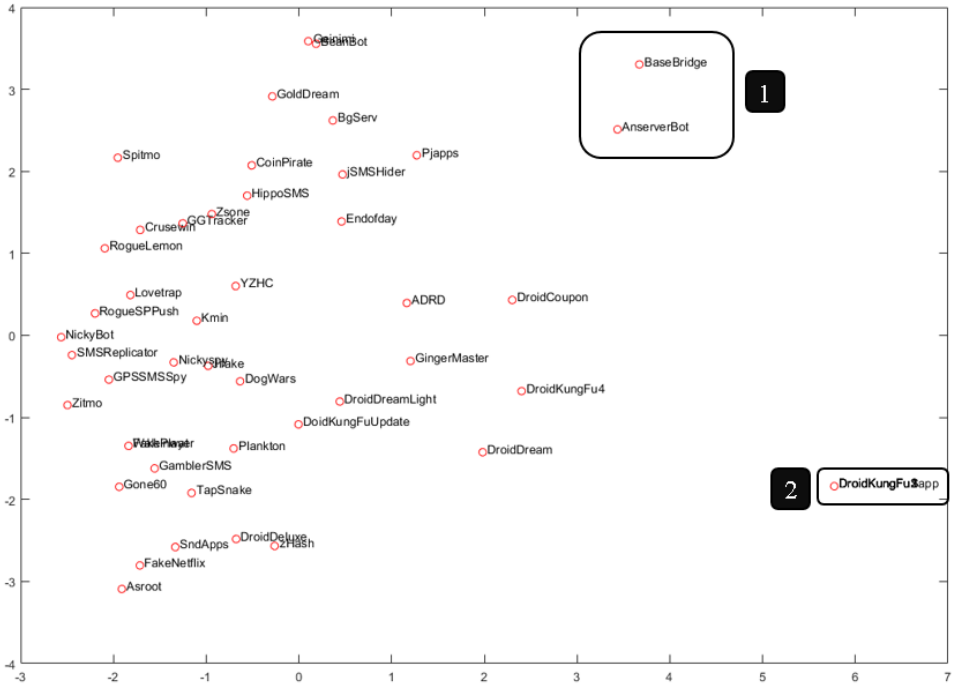


Figura 4.1: PCA projection of Malgenome families.

In Fig 1 it can be seen that most of the malware families are grouped in a main group (left side of the figure) while just a few families can be identified away from this cluster (groups 1 and 2). Group 1 gathers two families (*BaseBridge* and *AnserverBot*), that are the only two families in the dataset that combine repackaging and update installation. Group 2 gathers four families (*DroidKungFu1, DroidKungFu2, DroidKungFu3* and *DroidKungFuSapp*) that are the only ones in the dataset presenting the encrypted privilege escalation.

Additionally, this first projection let us identify that some families are projected at the very same place. By getting back to the data we have realized that these families take

the very same values for all the features. This is the case of *Walkinwat* and *FakePlayer* on the one hand and for *DroidKungFu1, DroidKungFu2, DroidKungFu3* and *DroidKungFuSapp* on the other hand. It means that, by taking into account the features in the analysed dataset, it will not be possible to distinguish *Walkinwat* from *FakePlayer* malware or any of the 4 mentioned variants of *DroidKungFu* malware.

**MLHL Projection**

Fig. 2 shows the MLHL projection of the analyzed data (two main components). MLHL projection shows the structure of the data in a way that a kind of ordering can be seen in the dataset. However, as it is more clearly shown in the CMLHL projection (Fig. 3), MLHL is not further described.

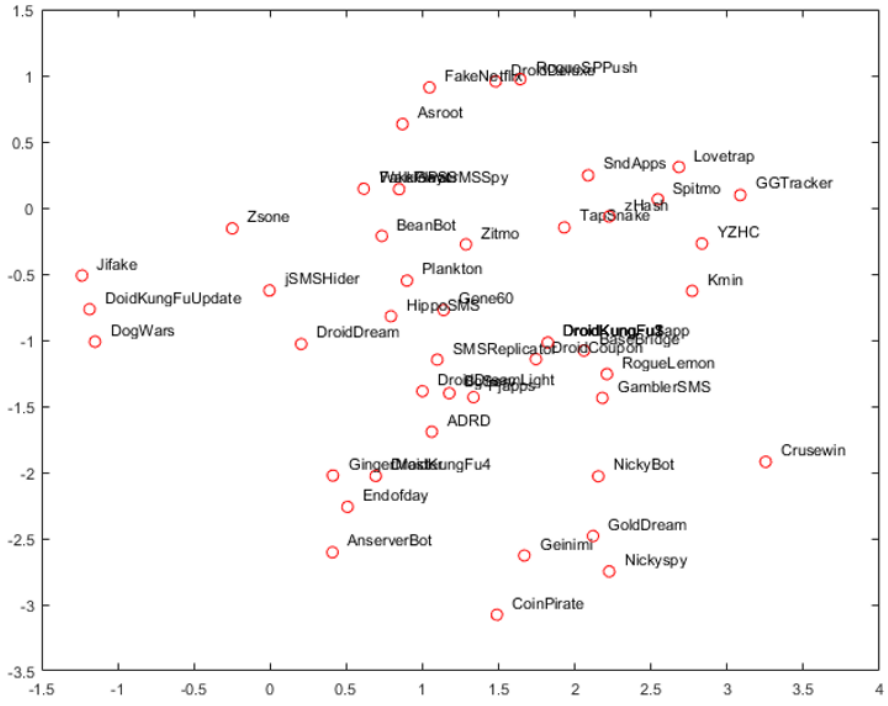


Figura 4.2: MLHL projection of Malgenome families.

The parameter values of the MLHL model for the projections shown in Fig. 2 are: Number of output dimensions: 3. Number of iterations: 100, learning rate: 0. 2872, p: 0.4852.

**CMLHL Projection**

When applying CMLHL to the analysed dataset, the projection (two main components) shown in Fig. 3 has been obtained. As expected, CMLHL obtained a sparser projection than MLHL and PCA, revealing the structure of the dataset in a clearer way.

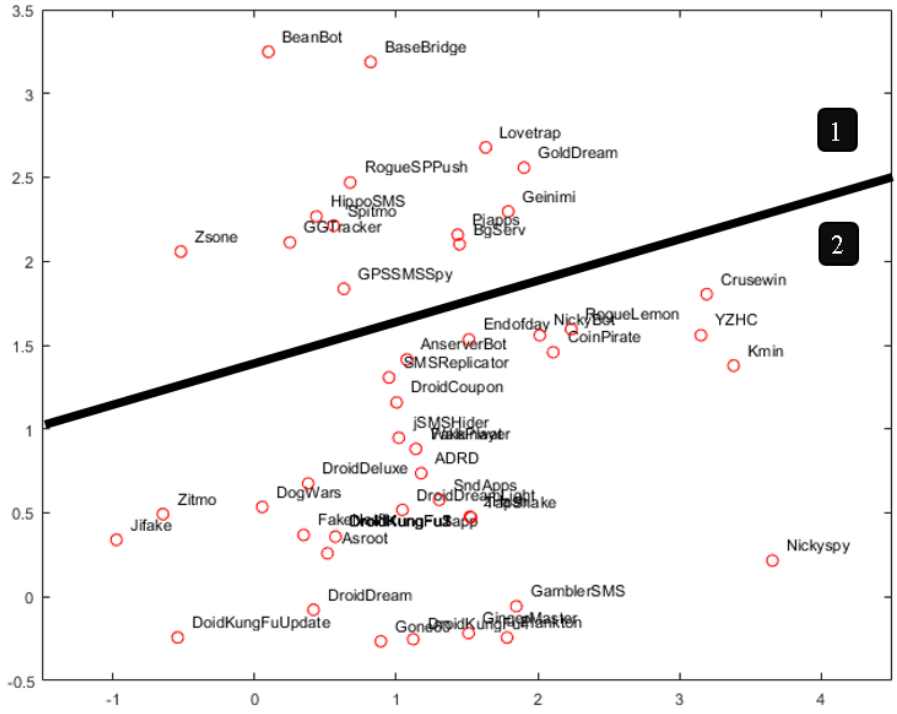


Figura 4.3: CMLHL projection of Malgenome families.

The parameter values of the CMLHL model for the projections shown in Fig. 3 are; Number of output dimensions: 3. Number of iterations: 100, learning rate: 0.0406, p: 1.92, $\tau$: 0.44056.

In Fig 3 it is easy to visually identify at least two main groups of data, labelled as 1 and 2. It has been checked that families in each one of these groups are similar in a certain way; group 1 gathers all the families with dangerous SMS activity, as "SMS activation" and "SMS financial charges" are present in all the families in this group. On the other hand, none of the families in this group present any of the following features: USB or "PKG activation", and "user-account personal information stealing". This group is also characterized by the almost complete absence of privilege escalation, as only one of those features (RATC/Zimperlich) is present in only one of the families (*BaseBridge*). Regarding group 2, none of the families in Group 2 present the feature "phone-call financial charges".

From a deeper analysis of such groups, some subgroups can be distinguished and are

identified in Fig. 4. Additionally, the families located in each one of these groups are listed in Table 2.

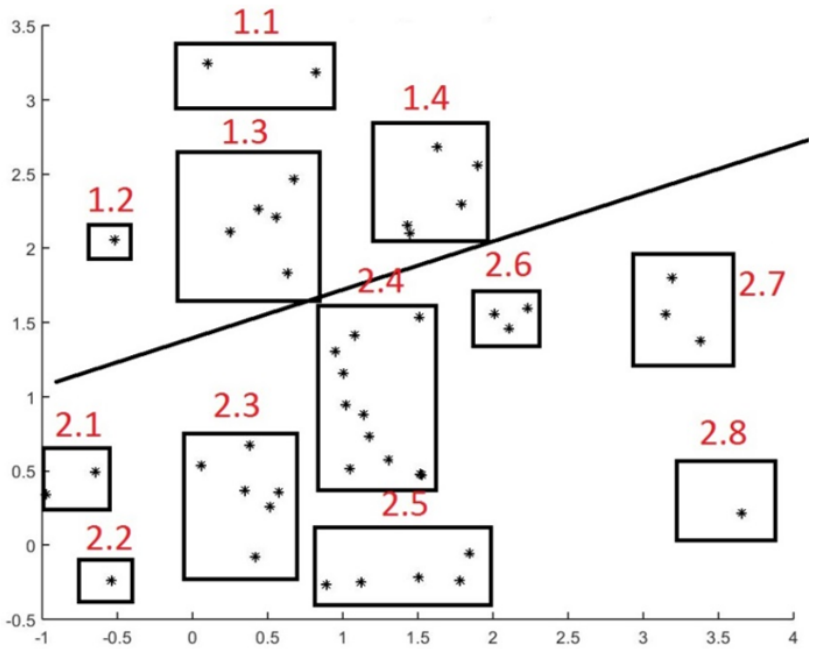


Figura 4.4: CMLHL projection of Malgenome families with identified subgroups.

Tabla 4.2: Families allocation to subgroups defined in CMLHL projection.

| **Group** | **Subgroup** | **Families** |
|---|---|---|
| 1 | 1.1 | *BaseBridge,BeanBot* |
| | 1.2 | *Zsone* |
| | 1.3 | *GGTracker, GPSSMSSpy, HippoSMS, RogueSPPush, Spitmo* |
| | 1.4 | *BgServ, Geinimi, GoldDream, Lovetrap, Pjapps* |
| 2 | 2.1 | *Jifake, Zitmo* |
| | 2.2 | *DroidKungFuUpdate* |
| | 2.3 | *Asroot, DogWars, DroidDeluxe, DroidDream, DroidKungFu1, DroidKungFu2, DroidKungFu3, DroidKungFuSapp, FakeNetflix* |
| | 2.4 | *ADRD, AnserverBot, DroidCoupon, DroidDreamLight, Endofday, FakePlayer,jSMSHider, SMSReplicator, SndApps, TapSnake, Walkinwat, zHash* |
| | 2.5 | *DroidKungFu4, GamblerSMS, GingerMaster, Gone60, Plankton* |
| | 2.6 | *CoinPirate, NickyBot, RogueLemon* |
| | 2.7 | *Crusewin, Kmin, YZHC* |
| | 2.8 | *Nickyspy* |

All the variants of *DroidKungFu* malware are located in the bottom-left side of the projection (groups 2.2, 2.3, and 2.5). *Jifake* and *Zitmo* are gathered in the same subgroup (2.1) as they are the only two families in group 2 presenting the drive-by download installation feature

**ISOMAP Projection**

In Fig. 5 it is shown the projections obtained by ISOMAP algorithm where each sample is labelled with the name of the family it belongs.

The parameter values of the ISOMAP model for the projection shown in Fig. 5 are:

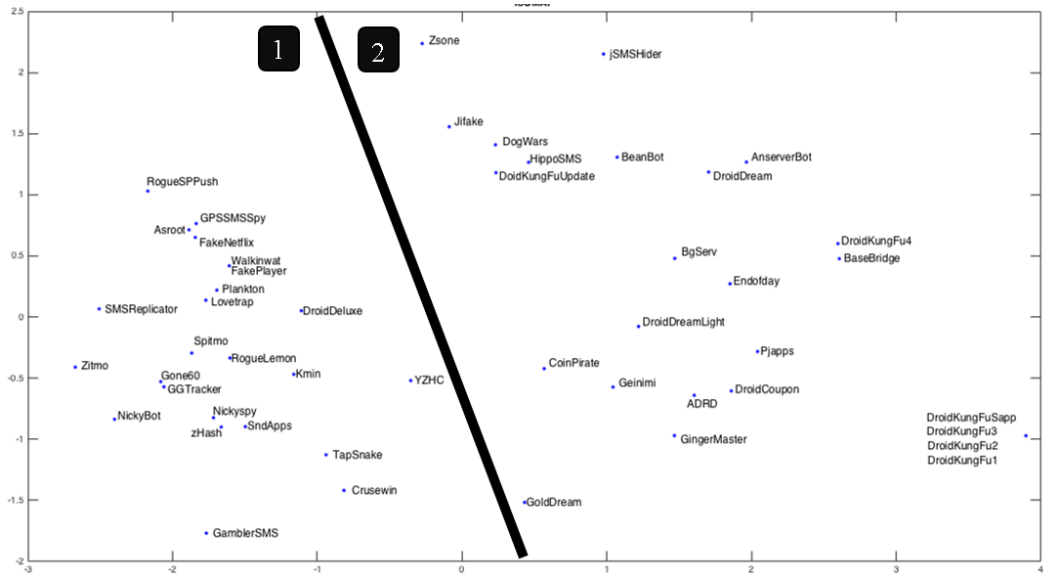


Figura 4.5: ISOMAP projection of Malgenome families.

number of neighbours: 10.

ISOMAP clearly visualize the internal dataset structure, with a main division into 2 groups (1 and 2 in Fig. 5). For a deeper analysis, these two main groups are split in different subgroups as shown in Fig. 6.

From groups in Fig 5, it can be highlighted that group 1 contains Malgenome families that present "standalone" but not "repackaging" installation features (see Table 3). However, in case of group 2, none of its samples present "standalone installation" feature and all of them present the "repackaging installation" feature.

As shown in Fig. 6, group 1 is divided in 2 subgroups (G1a and G1b), where G1a gathers families that do not present the "BOOT activation" feature as opposed to G1b, where its samples present this "BOOT activation" feature. Similarly, G2 is clearly

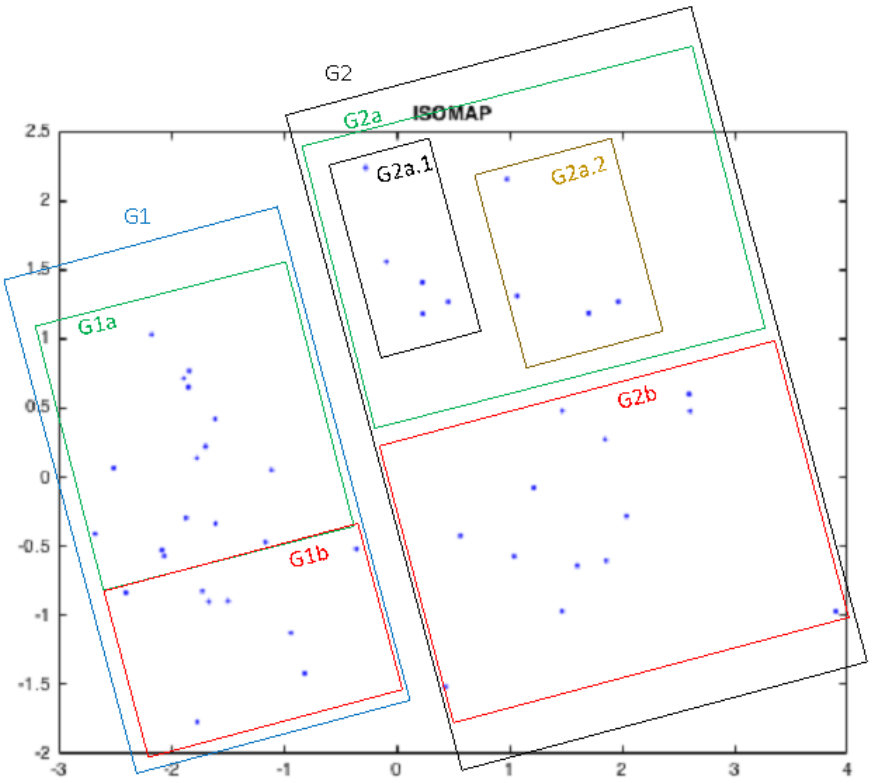


Figura 4.6: ISOMAP projections of Malgenome families with identified subgroups.

divided into 2 subgroups (G2a and G2b), with analogous characteristics to samples in G1a and G2a respectively ("BOOT activation" for G2a and the opposite for G2b). Finally, G2a presents samples with dangerous activity of "NET remote control" (group G2a.1), and samples without such dangerous feature (group G2a.2).

Table 3 shows the Malgenome families contained in each one of the identified groups, and the features characterizing all the families in that group.

Tabla 4.3: Families allocation to subgroups defined in ISOMAP projection.

| **Group** | **Subgroup** | **Families and features** |
|---|---|---|
| G1 | G1a | RogueSPPush, GPSSMSSpy, Asroot, FakeNetflix, Walkinwat, FakePlayer, Plankton, SMSReplicator, Lovetrap, DroidDeluxe, Spitmo, Zitmo, Spitmo, RogueLemon, Gone60, GGTracker, Kmin<br>**Present features: 4 (standalone installation) and 5 (BOOT activation)**<br>**Not-present feature: 1(repackaging installation)** |
| | G1b | NickyBot, Nickyspy, zHash, SndApps, YZHC, TapSnake, Crusewin, GamblerSMS<br>**Present features: 4 (standalone installation)**<br>**Not-present feature:1(repackaging installation) and 5 (BOOT activation)** |

| Group | Subgroup | | Families and features |
|---|---|---|---|
| G2 | G2a | G2a.1 | Zsone, Jifake, DogWars, HippoSMS, DoidKungFuUpdate<br>**Present features: 1 (repackaging installation) and 5 (BOOT activation)**<br>**Not-present feature: 4 (standalone installation) and 19 (NET remote control)** |
| | | G2a.2 | jSMSHider, BeanBot, AnserverBot, DroidDream<br>**Present features: 1 (repackaging installation), 5 (BOOT activation) and 19 (NET remote control)**<br>**Not-present feature: 4 (standalone installation)** |
| | G2b | | DroidKungFu4, BaseBridge, BgServ, Endofday, DroipDreamLight, Pjapps, DroidCoupon, CoinPirate, Geinimi, ADRD, GingerMaster, DroidKungFuSapp, DroidKungFu3, DroidKungFu2, DroidKungFu1<br>**Present features: 1 (repackaging installation) and 19 (NET remote control)**<br>**Not-present feature: 4 (standalone installation), and 5 (BOOT activation)** |

### CCA Projection

Fig. 7 presents the projection obtained by CCA of Malgenome families, where it can be seen that a clear internal structure of the dataset can not be identified, and malware families can not be clearly gathered in groups, as it happened in previous results.

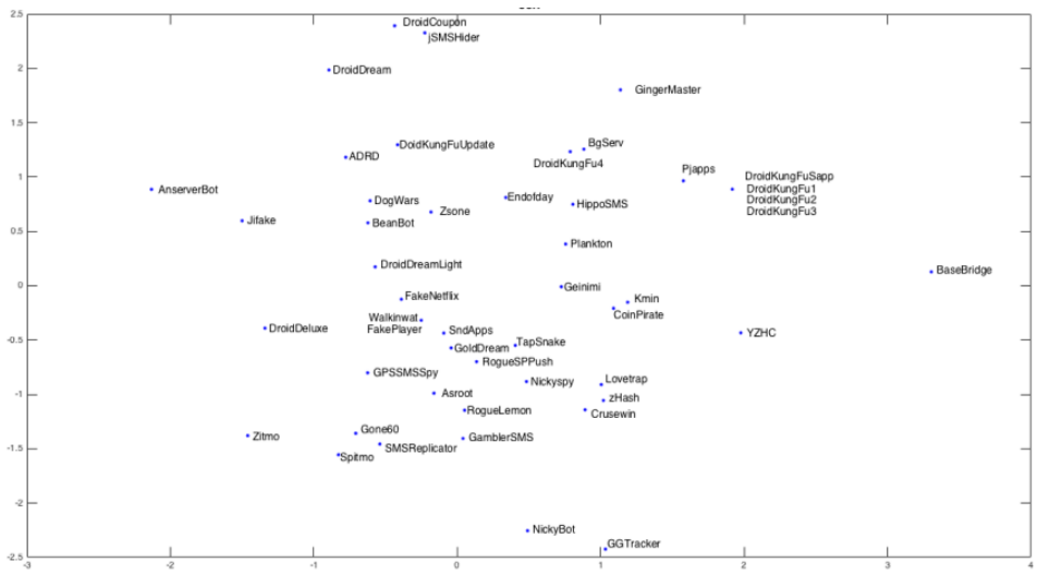


Figura 4.7: CCA projection of Malgenome families.

Different combinations of values were tested for the parameters of the CCA model. The best projection obtained is the one shown in Fig. 7, that was generated with 1,000 epochs, alpha=0.5 and lambda=1.5152.

**SOM results**

Finally, SOM has been also applied to the Malgenome dataset and the obtained U-matrix is shown in Fig. 4.8. Each one of the neurons in the map has been labelled with the names of the malware families to which the neuron responds. From this figure, and according to the inter-neuron distances, neurons in the map could be easily split in two main groups (G1 and G2). At the same time, G1 could be divided in two subgroups (G1a and G1b), and G1a could also be divided into three subgroups as neuron distances are high between them (blue color means high distance in Fig. 4.8). In the case of G2, authors believe that it can not be divided into subgroups, as neuron distances within G2 are quite small, so it can not be said that families in this group (G2) are very different.

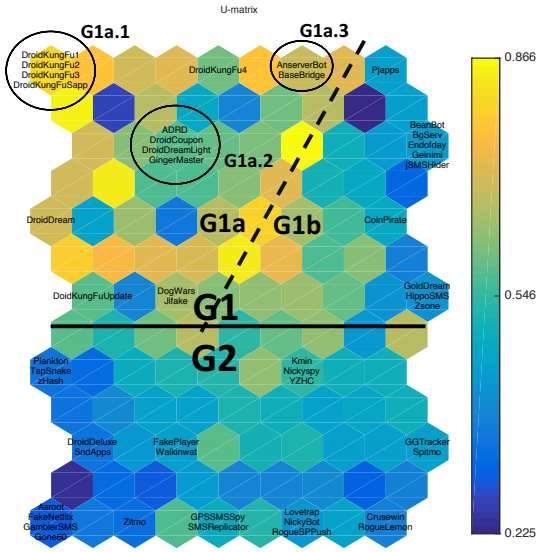


Figura 4.8: SOM U-matrix for Malgenome families with identified groups.

The parameter values of the SOM model for the mapping shown in Fig. 8 are; map size: [7, 5], lattice: hexagonal, neighbourhood function: Gaussian. On the other hand, some metrics about the obtained mapping are: quantization error = 1.1, and topographic error = 0.0. In general terms, it can be said that the "repackaging installation" and "standalone installation" features are the only ones that let distributing samples in groups G1 and G2. In the case of subgroups G1a and G1b, it is the presence of the "SMS financial charges" feature what characterize malware families in each one of them.

Table 4.4 shows the Malgenome families contained in each one of the identified groups by the SOM network, and the features characterizing all the families in that group.

General information (present and not-present features) of a group (i.e. group 1) is applicable to all malware families contained in its subgroups (i.e. subgroups G1a1, G1b, etc.).

Tabla 4.4: Families allocation to subgroups defined in SOM u-matrix.

| | |
|---|---|
| **G1** | |
| **Present feature: 1 (repackaging installation)** | |
| **Not-present feature: 4 (standalone installation)** | |
| **G1a** | |
| **Present feature: 22 (SMS financial charges)** | |
| G1a1 | DroidKungFu1, DroidKungFu2, DroidKungFu3, DroidKungFuSapp<br>**Present features: 11 (BATT activation), 14 (exploid privilege escalation), 18 (encrypted privilege escalation)**<br>**Not-present feature: 2 (update installation), 24 (SMS personal information stealing)** |
| G1a2 | DroidCoupon, DroidDreamLight, GingerMaster, ADRD<br>**Not-present feature: 11 (BATT activation), 14 (exploid privilege escalation), 18 ( encrypted privilege escalation), 2 (update insatalltion), 24 (SMS personal information stealing)** |
| G1a3 | AnserverBot, BAseBridge<br>**Present features: 11 (BATT activation), 2 (update installation), 24 (SMS personal information stealing)**<br>**Not-present feature: 14 (exploid privilege escalation), 18 (encrypted privilege escalation)** |
| Others samples | DogWars, DroidkungFuUpdate, DroidDream, DroidKungFu4, Jifake |
| **G1b** | |
| **Not present features: 22 (SMS financial charges)** | |
| BeanBot, BgServ, CoinPirate, Endofday, Geinimi, GoldDream, HippoSMS, jSMSHider, Pjapps, Zsone | |
| **G2** | |
| **Present features: 4 (standalone installation)** | |
| **Not-present feature: 1 (repackaging installation)** | |
| Asroot, Crusewin, DroidDeluxe, FakeNetflix, FakePlayer, GamblerSMS, GGTracker, Gone60, GPSSMSSpy, Kmin, Lovetrap, NickyBot, Nickyspy, Plankton, RogueLemon, RogueSPPush, SMSReplicator, SndApps, Spitmo, TapSnake, Walkinwat, YZHC, zHash, Zitmo | |

## 4.4. Conclusions

From the results shown in section 3, it can be concluded that dimensionality reduction techniques are an interesting proposal to visually analyse the structure of a high-dimensionality dataset in general terms. More specifically, when studying Android malware families, this kind of techniques let us gain deep knowledge about the nature of such app families. Thanks to the obtained projections, similarities and differences of the studied families are identified.

From the extensive set of applied DRTs, PCA, MLHL and CCA failed in generating an informative visualization of samples by reducing the dimensionality of them to 2D. On the other hand and generally speaking, it can be said that the DRTs that group malware families, are able to do that in a way consistent with the seminal characterization of Malgenome dataset [7]. More precisely, it is worth mentioning that installation features (repackaging and standalone) have been identified by ISOMAP and SOM as the most important ones for a general characterization of Android malware families (see section 3 for further details). It is an important result as repackaging is one of the most common techniques applied to hide malware (86 % of the malware apps in the original dataset were repackaged versions of different legitimate apps including paid apps, popular game apps, powerful utility apps, etc. [6]).
In a complementary way, CMLHL identified some activation (SMS, USB, and PKG) and financial charges (SMS, and phone call) as the most important ones for such a task. Knowledge generated by the application of DRTs could be applied to improve the detection rate of Android Malware at different stages (markets, devices, etc.) thanks to the characterization of the different families.
As a final conclusion, it can be said that the identification and characterization of Android malware is still and open challenge that requires great efforts to be devoted in coming years.

## 4.5. Future work

As future work it is planned to apply new DTRs models and compare them with other supervised algorithms such as decision threes in order to gain deep knowledge of the dataset. It will also be analysed other datasets related to cybersecurity applying the same approach followed in this research in order to generalize to other datasets the proposed method.

## Bibliografía

[1] A. Altaher, "An improved android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (EHNFC) and permission-based features," *Neural Computing and Applications*, vol. 28, no. 12, pp. 4147–4157, 2017.

[2] AppBrain. Android operating system statistics. [Online]. Available: https://www.appbrain.com/stats/stats-index

[3] M. Proyect. Android malware genome project. [Online]. Available: http://www.malgenomeproject.org

[4] S. Arshad, M. A. Shah, A. Khan, and M. Ahmed, "Android malware detection & protection: a survey," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, pp. 463–475, 2016.

[5] D. Atienza, Á. Herrero, and E. Corchado, "Neural analysis of http traffic for web attack detection," in *Computational Intelligence in Security for Information Systems Conference*. Springer, 2015, pp. 201–212.

[6] L. Sáiz, A. Pérez, Á. Herrero, and E. Corchado, "Analyzing key factors of human resources management," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2011, pp. 463–473.

[7] I. J. Machón González, H. López García, and J. L. Calvo Rolle, "Neuro-robust controller for non-linear systems (controlador neurorobusto para sistemas no lineales)," *Dyna*, 2011.

[8] L. Cen, C. S. Gates, L. Si, and N. Li, "A probabilistic discriminative model for android malware detection with decompiled source code," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 400–412, 2014.

[9] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. I–I.

[10] N. Chen, B. Ribeiro, A. Vieira, and A. Chen, "Clustering and visualization of bankruptcy trajectory using self-organizing map," *Expert Systems with Applications*, vol. 40, no. 1, pp. 385–393, 2013.

[11] G. Cirrincione, J. Hérault, and V. Randazzo, "The on-line curvilinear component analysis (oncca) for real-time data reduction," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[12] E. Corchado and C. Fyfe, "Connectionist techniques for the identification and suppression of interfering underlying factors," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 08, pp. 1447–1466, 2003.

[13] E. Corchado, Á. Herrero, and J. M. Sáiz, "Testing cab-ids through mutations: on the identification of network scans," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2006, pp. 433–441.

[14] E. Corchado and Á. Herrero, "Neural visualization of network traffic data for intrusion detection," *Applied Soft Computing*, vol. 11, no. 2, pp. 2042–2056, 2011.

[15] E. Corchado, D. MacDonald, and C. Fyfe, "Maximum and minimum likelihood hebbian learning for exploratory projection pursuit," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 203–225, 2004.

[16] C. C. Turrado, M. d. C. M. López, F. S. Lasheras, B. A. R. Gómez, J. L. C. Rollé, and F. J. d. C. Juez, "Missing data imputation of solar radiation data under different atmospheric conditions," *Sensors*, vol. 14, no. 11, pp. 20 382–20 399, 2014.

[17] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on neural networks*, vol. 8, no. 1, pp. 148–154, 1997.

[18] J. J. Fuertes, M. Domínguez, P. Reguera, M. A. Prada, I. Díaz, and A. A. Cuadrado, "Visual dynamic model based on self-organizing maps for supervision and fault detection in industrial processes," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 1, pp. 8–17, 2010.

[19] C. Fyfe, "A neural network for pca and beyond," *Neural Processing Letters*, vol. 6, no. 1-2, pp. 33–41, 1997.

[20] C. Fyfe, D. R. McGregor, and R. Baddeley, *Exploratory projection pursuit: an artificial neural network approach*. Department of Computer Science, University of Strathclyde, 1994.

[21] R. F. Garcia, J. L. C. Rolle, M. R. Gomez, and A. D. Catoira, "Expert condition monitoring on hydrostatic self-levitating bearings," *Expert Systems with Applications*, vol. 40, no. 8, pp. 2975–2984, 2013.

[22] H. Fakhouri, L. Cherrat, and M. Ezziyyani, "Towards a new approach to improve the classification accuracy of the kohonen's self-organizing map during learning process."

[23] A. Herrero, U. Zurutuza, and E. Corchado, "A neural-visualization ids for honeynet data," *International Journal of Neural Systems*, vol. 22, no. 02, p. 1250005, 2012.

[24] S. Hou and P. D. Wentzell, "Re-centered kurtosis as a projection pursuit index for multivariate data analysis," *Journal of Chemometrics*, vol. 28, no. 5, pp. 370–384, 2014.

[25] J.-w. Jang, J. Yun, A. Mohaisen, J. Woo, and H. K. Kim, "Detecting and classifying method based on similarity matching of android malware behavior with profile," *SpringerPlus*, vol. 5, no. 1, p. 273, 2016.

[26] T. Kohonen, "Essentials of the self-organizing map," *Neural networks*, vol. 37, pp. 52–65, 2013.

[27] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1-3, pp. 1–6, 1998.

[28] Malwarebytes, "Malwarebytes labs report: Cybercrime tactics and techniques q3 2017," Tech. Rep., 2017.

[29] E. Mohebi and A. Bagirov, "Constrained self organizing maps for data clusters visualization," *Neural Processing Letters*, vol. 43, no. 3, pp. 849–869, 2016.

[30] V. Moonsamy, J. Rong, and S. Liu, "Mining permission patterns for contrasting clean and malicious android applications," *Future Generation Computer Systems*, vol. 36, pp. 122–132, 2014.

[31] E. Oja, "Principal components, minor components, and linear neural networks," *Neural networks*, vol. 5, no. 6, pp. 927–935, 1992.

[32] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," *Expert systems with applications*, vol. 36, no. 1, pp. 2–17, 2009.

[33] W. Park, K.-H. Lee, K.-S. Cho, and W. Ryu, "Analyzing and detecting method of android malware via disassembling and visualization," in *2014 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2014, pp. 817–818.

[34] A. Paturi, M. Cherukuri, J. Donahue, and S. Mukkamala, "Mobile malware visual analytics and similarities of attack toolkits (malware gene analysis)," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, 2013, pp. 149–154.

[35] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[36] C. I. Pinzón, J. F. De Paz, A. Herrero, E. Corchado, J. Bajo, and J. M. Corchado, "idmas-sql: intrusion detection based on mas to detect and block sql injection through data mining," *Information Sciences*, vol. 231, pp. 15–31, 2013.

[37] C. Pinzón, J. F. De Paz, J. Bajo, Á. Herrero, and E. Corchado, "Aiida-sql: An adaptive intelligent intrusion detector agent for detecting sql injection attacks," in *2010 10th International Conference on Hybrid Intelligent Systems*. IEEE, 2010, pp. 73–78.

[38] H. Quintián and E. Corchado, "Beta hebbian learning as a new method for exploratory projection pursuit," *International journal of neural systems*, vol. 27, no. 06, p. 1750024, 2017.

[39] H. Quintián and E. Corchado, "Beta scale invariant map," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 218–235, 2017.

[40] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

[41] R. Sánchez, Á. Herrero, and E. Corchado, "Visualization and clustering for snmp intrusion detection," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 505–532, 2013.

[42] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, J. Nieves, P. G. Bringas, and G. Álvarez Marañón, "Mama: manifest analysis for malware detection in android," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 469–488, 2013.

[43] Statista. Smartphone sales by os worldwide 2009-2017. [Online]. Available: https://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operatingsystem/

[44] O. Somarriba, U. Zurutuza, R. Uribeetxeberria, L. Delosières, and S. Nadjm-Tehrani, "Detection and visualization of android malware behavior," *Journal of Electrical and Computer Engineering*, vol. 2016, 2016.

[45] SOPHOSLABS, "Sophoslabs 2018 malware forecast," Tech. Rep., 2017.

[46] P. Teufl, M. Ferk, A. Fitzek, D. Hein, S. Kraxberger, and C. Orthacker, "Malware detection by applying knowledge discovery processes to application metadata on the android market (google play)," *Security and communication networks*, vol. 9, no. 5, pp. 389–419, 2016.

[47] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner, "A survey of visualization systems for malware analysis," in *Eurographics Conference on Visualization (EuroVis)*, 2015, pp. 105–125.

[48] Y. Wu, T. K. Doyle, and C. Fyfe, "Multi-layer topology preserving mapping for k-means clustering," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2011, pp. 84–91.

[49] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in *2012 IEEE symposium on security and privacy*. IEEE, 2012, pp. 95–109.

[50] H.-J. Zhu, Z.-H. You, Z.-X. Zhu, W.-L. Shi, X. Chen, and L. Cheng, "Droiddet: effective and robust detection of android malware using static analysis along with rotation forest model," *Neurocomputing*, vol. 272, pp. 638–646, 2018.

# CAPÍTULO 5

# Delving into Android Malware Families with a Novel Neural Projection Method Dimensionality Reduction Techniques

En este capítulo se presenta el segundo de los artículos utilizado para la defensa de la tesis por compendio de artículos. A continuación se presentan los datos básicos del mismo, y después de incluirá todo el contenido del artículo copiado del artículo publicado. Se ha decidido no incluir el artículo original para mantener una uniformidad en la presentación de la tesis, pero se puede consultar a través del *doi* correspondiente.

Datos generales del artículo:

**Autores:**
Rafael Vega Vega[1], Héctor Quintián[1], Carlos Cambra[2], Nuño Basurto[2], Álvaro Herrero[2] and José Luis Calvo-Rolle[1,3]
**Afiliaciones:**
[1] Department of Industrial Engineering, University of A Coruña, A Coruña, Spain
Avda. 19 de febrero S/N 15.495, Ferrol - Coruña, Spain
rafael.alejandro.vega.vega@udc.es, hector.quintian@udc.es, jlcalvo@udc.es
[2] Grupo de Inteligencia Computacional Aplicada (GICAP),
Departamento de Ingeniería Civil, Escuela Politécnica Superior, Universidad de Burgos
Av. de Cantabria s/n, 09006, Burgos, Spain
ccbaseca@ubu.es,nbasurto@ubu.es,ahcosio@ubu.es
[3] Research Institute of Applied Sciences in Cybersecurity (RIASC)
**Título:** Delving into Android Malware Families with a Novel Neural Projection Meth -od
**Revista:** Complexity
**Publicación:** Vol. 2019
**Páginas:** 1–10
**Editorial:** Wiley-Hindawi
**Año de publicación:** 2019
**DOI:** 10.1155/2019/6101697
**Factor de impacto JCR 2019:** 2,462
**Factor de impacto JCR 2019 (5 años):** 2,474
**Disciplina *Multidisciplinary sciences (SCIE)* 2019:** 31/71 - Q2
**Disciplina *Mathematics, interdisciplinary applications (SCIE)* 2019:** 28/106 - Q2

## Abstract

Present research proposes the application of unsupervised and supervised machine-learning techniques to characterize Android malware families. More precisely, a novel unsupervised neural-projection method for dimensionality-reduction, namely Beta Hebbian Learning (BHL), is applied to visually analyze such malware. Additionally, well-known supervised Decision Trees (DTs) are also applied for the first time in order to improve characterization of such families and compare the original features that are identified as the most important ones. The proposed techniques are validated when facing real-life Android malware data by means of the well-known and publicly-available Malgenome dataset. Obtained results supports the proposed approach, confirming the validity of BHL and DTs to gain deep knowledge on Android malware.

**keywords:** Machine Learning, Neural Networks, Dimensionality-reduction, Decision Trees, Android Malware

## 5.1. Introduction and Previous Work

Undoubtedly, smartphones are one of the emerging technologies that have revolutionized the use of computing systems. From the very beginning (late 1990s), more and more smartphones are sold every year and it is expected that the number of smartphone users pass the 2.7 billion mark by 2019 [1]. Although there is a variety of operating systems for such devices, Google's Android is the most widely-used one [1] and consequently, the number of Android users has permanently increased. Concurrently, the number of Android apps strongly increased in the last years but it started to decline from 3.6 million in March, 2017 (highest value) to 2.6 million in September, 2018 [2].

From the security standpoint, one of the main problems of smartphone apps is malware, that is included in software in general and in these apps in particular. Furthermore, "users of mobile devices are increasingly subject to malicious activity pushing malware apps" [3]. It is true that some effort has been devoted by Google to remove and prevent malicious apps from Google Play Market, but malware is still there [3]. Moreover, malware Android apps are increasing; in the third trimester of 2018 there has been an increase of 1.7 million detections [4].

As it can be seen, privacy and security of smartphones still are open challenges [5] and many researchers are working on this topic. To better fight against malware and be able to develop an effective solution, understanding it and its nature, is required [6]. In keeping with this idea, present paper proposes getting deeper knowledge about Android malware for its better detection. More precisely, both supervised (Decision Trees) and unsupervised (Neural Projection Method) machine-learning techniques are applied to increase our knowledge about the main families of Android malware. In order to validate the proposed techniques, they are applied to the well-known Malgenome dataset [7], that is open and real-life.

This pioneering work on collecting Android malware found some interesting statistics [6] motivating further analysis of malware: 36.7 % of the collected apps leverage root-level exploits to fully compromise the security of the smartphone and more than 90 % of the apps tried to turn the smartphone into a botnet controlled through network or short messages.

To improve present knowledge of Android malware families, a novel neural-projection technique from the family of Exploratory Projection Pursuit (EPP) techniques, named Beta Hebbian Learning (BHL) [8] is applied. Obtained results are then compared to those from several different Decision Trees (DTs) [9] when trying to predict the malware family from apps features.

Each app (data sample) that was collected for the Malgenome dataset is defined as a set of certain features using a binary representation. Apps were grouped according to the family they belong to, and features were recalculated for the whole family, taking into account which features were present in the given apps. The generated high-dimensional space is then analysed by means of BHL in order to reveal the inner structure of the dataset. Obtained projections are consequently scrutinized to get further knowledge about the app features that define the organization of the data in different groups and subgroups. For comparison purposes, DTs have been additionally generated on the same features set, in order to know the features that better discriminate between the different malware families.

A variety of problems have been addressed by artificial neural networks in recent decades [10, 11, 12, 13, 14]. More precisely, neural projection models have been previously applied to a wide variety of security datasets, including network traffic [15, 16], SQL code [17, 18], and HTTP traffic [19]. Similarly, from a more general perspective, different machine learning solutions have been proposed to differentiate between

legitimate and malicious apps [20, 21, 22].

Visualization techniques have been previously applied to this problem of analyzing malware [23, 24, 25, 26, 27, 28, 29]. However, few dimensionality-reduction techniques have been applied to Android apps in order to detect malware; Pythagoras tree fractal visualization is proposed in [25], being all apps scattered, as leaves in the tree. Graphs for deciding about malicious apps by depicting lists of malicious methods, needless permissions and malicious strings were proposed in [26]. Biclustering on permission information was used to generate a visualization in [27], while behavior-related dendrograms are generated out of malware traces in [28]. In the later, different pieces of information are analysed, including nodes related to the package name of the application, the Android components that has called the API call, and the names of functions and methods invoked by the application. Differentiating from previous work, in present paper, a novel neural projection technique is applied for the first time to the characterization of Android malware [8, 24, 30]. Apps are not analysed one by one, but family-level is considered instead. Additionally, DTs are applied for the first time in order to improve characterization of such families.

The rest of this paper is organized as follows, initially BHL and DTs are presented and the analyzed dataset is described in the following section. Then, the proposed experiments are introduced and the obtained results are analyzed in Section 3. Finally, conclusions and future work are presented in Section 4 of the paper.

## 5.2. Materials and Methods

In present research, the EPP BHL algorithm [8] has been applied to a dataset of malware families with the aim of identifying the internal structure of such dataset and finding families of malware with similar characteristics. The obtained results have been compared with a well known prediction algorithm (DT) [9] to validate the BHL results regarding the most relevant features to briefly characterize Malware families.

### 5.2.1. Beta Hebbian Learning

The Beta Hebbian Learning technique (BHL) [8] is an unsupervised neural network from the family of EPP that employs the Beta distribution to update its learning rule and fit the Probability Density Function (PDF) of the residual with the distribution of a given dataset.

Thus, if the PDF of the residuals is known, the optimal cost function can be determined. By using $B(\alpha, \beta)$ parameters of the Beta distribution, the residual ($e$) can be drawn with the following PDF (Eq. 5.1):

$$p(e) = e^{\alpha-1}(1-e)^{\beta-1} = (x - Wy)^{\alpha-1}(1 - x + Wy)^{\beta-1} \tag{5.1}$$

Where $\alpha$ and $\beta$ are used to adjust the shape of the PDF of the Beta distribution, $x$ is the input of the network, $e$ is the residual, $W$ is the weight matrix, and $y$ is the output of the network.

Then, by using Eq. 5.2, gradient descent is performed to maximize the likelihood of the weights:

$$\begin{aligned}\frac{\partial p}{\partial W} = (e_j^{\alpha-2}(1-e_j)^{\beta-2}(-(\alpha-1)(1-e_j) + e_j(\beta-1))) = \\ (e_j^{\alpha-2}(1-e_j)^{\beta-2}(1-\alpha+e_j(\alpha+\beta-2)))\end{aligned} \tag{5.2}$$

In the case of BHL, the learning rule allows to fit the PDF of the residual, by maximizing the likelihood of such residual with the current distribution.
Therefore, the neural architecture for BHL is defined as follows:

$$Feedforward : y_i = \sum_{j=1}^{N} W_{ij}x_j, \forall i \tag{5.3}$$

$$Feedback : e_j = x_j - \sum_{i=1}^{M} W_{ij}y_i \tag{5.4}$$

$$Weightsupdate : \Delta W_{ij} = \eta(e_j^{\alpha-2}(1-e_j)^{\beta-2}(1-\alpha+e_j(\alpha+\beta-2)))y_i \tag{5.5}$$

### 5.2.2. Decision Trees

Decision Trees (DTs) [9] are machine-learning algorithms widely used for prediction, that have proved their benefits in several real applications. They can be categorized as supervised non-parametric inductive learning techniques. They are based on the construction of diagrams from a dataset, in a similar way that prediction systems based on rules, which serve to represent and categorize a series of conditions that occur repeatedly for the solution of a problem.

The main objective of a classification DT is to divide a dataset into groups of samples as similar as possible in relation to one of the features. They are made of three main elements: root node (contains all samples of the dataset), decision nodes (represent a decision or rule) and leaf nodes (final label). A dataset is then classified based on sub-divisions of the DT nodes to reach one of the final (leaf) nodes whose label corresponds to a class (Figure 5.1).

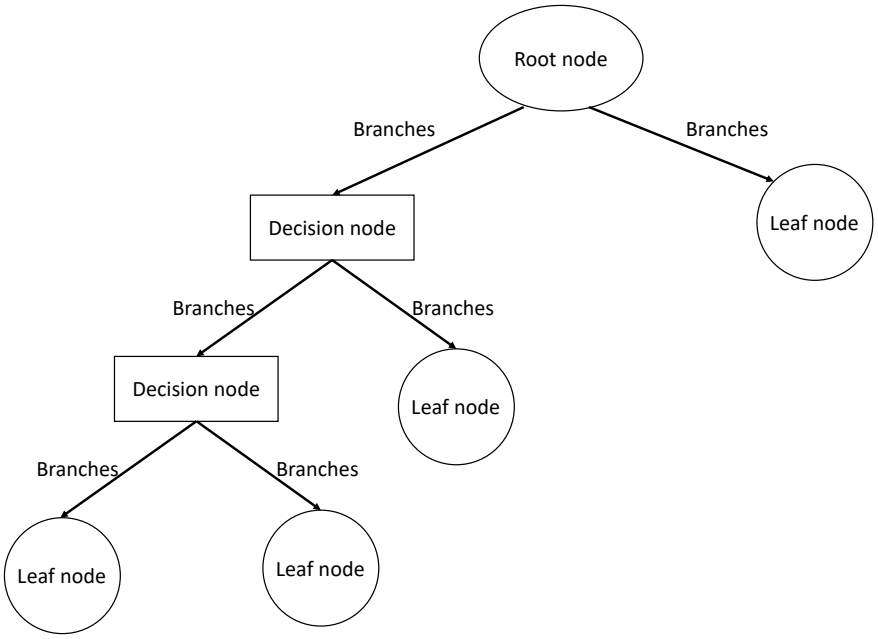


Figura 5.1: Structure of decision trees

Several algorithms have been proposed so far to build DTs and their efficiency has been proved. The most notable ones [31] are: ID3 (Iterative Dichotomiser 3), C4.5 (successor of ID3), CART (Classification And Regression Tree), CHAID (CHi-squared Automatic Interaction Detector), MARS, and Conditional Inference Trees. Among all of them, CART has been selected in present work due to two main reasons: the binary nature of the dataset and the main objective of the study (to identify the most relevant features of the dataset) [31].

CART

The Classification And Regression Tree (CART) [9] is a binary tree, so each decision node has two binary branches determined by a splinting function obtained by processing variance function. In order to build the tree, this CART algorithm takes 4 main steps [9]:

1. Build the decision tree splitting nodes according to a given function.
2. Finish tree construction once the learning fits the stop criteria.
3. Pruning the tree to avoid over-fitting.
4. Select the best tree after pruning process.

Originally, the splitting function used by CART is the Gini Index (equation 5.6).

$$Gini(S) = 1 - \sum_{i=1}^{n} p_i^2 \tag{5.6}$$

where $S$ is the dataset, $n$ is the number of classes in the dataset, and $p$ is the probability of different classes. Therefore, a Gini index of *0* means a *100 %* accuracy in predicting the class.

For comparison purposes, two other splitting functions have been applied in present paper: Deviance (equation 5.7) and Twoing (equation 5.8).

$$Deviance(S) = -\sum_{i=1}^{n} p_i log_2 p_i \tag{5.7}$$

Twoing is an splitting function different from Gini and Deviance. Being $L_i$ and $R_i$ the fraction of members of class $i$ in the left and right child nodes after a split respectively. $P_L$ and $P_R$ are the fractions of observations that split to the left and right respectively. Therefore, the function to be maximized is the one in equation 5.8.

$$P_L P_R (\sum_{i=1}^{n} |L_i - Ri|)^2 \tag{5.8}$$

On the other hand, in standard CART algorithm, the split feature that is selected for a decision node is the one that maximizes the split-criterion gain. Once again, for a more comprehensive comparison, two other criteria have been applied for selecting split features: Curvature [32] and Interaction [33]. These criteria can be defined as follows:

- Curvature: it is based on the null hypothesis of unassociated two features. With this criteria, the best split predictor feature is the one that minimizes the significant $p$-values of curvature tests between each feature and the response variable. Such a selection is robust to the number of levels in individual features.
- Interaction: it is based on the null hypothesis of no interaction between the label and the predictor features. Therefore, for deep decision trees, standard CART tends to miss important interactions between pairs of features when there are also many other less important features. By means of this criterion, the detection of such important interactions is improved.

### 5.2.3. Malgenome Dataset

The dataset used in this research has been obtained from the Android Malware Genome Project [7], which consist on 1260 Android malware samples grouped in a total of 49 malware families. It was collected from August 2010 to October 2011 and still is a standard benchmark dataset for Android Malware.

This dataset contains malware apps installed in user phones and based on 3 main attack strategies: repackaging, update attack, and drive-by download. Samples of this dataset were manually classified based on different aspects such as installation and activation mechanisms and malicious payloads nature. Collected malware was split in families, that were obtained “by carefully examining the related security announcements,

threat reports, and blog contents from existing mobile antivirus companies and active researchers as exhaustively as possible and diligently requesting malware samples from them or actively crawling from existing official and alternative Android Markets"[6].

The different families present in the dataset are: ADRD, AnserverBot, Asroot, BaseBridge, BeanBot, BgServ, CoinPirate, Crusewin, DogWars, DroidCoupon, DroidDeluxe, DroidDream, DroidDreamLight, DroidKungFu1, DroidKungFu2, DroidKungFu3, DroidKungFu4, DroidKungFuSapp, DoidKungFuUpdate, Endofday, FakeNetflix, FakePlayer, GamblerSMS, Geinimi, GGTracker, GingerMaster, GoldDream, Gone60, GPSSMSSpy, HippoSMS, Jifake, jSMSHider, Kmin, Lovetrap, NickyBot, Nickyspy, Pjapps, Plankton, RogueLemon, RogueSPPush, SMSReplicator, SndApps, Spitmo, TapSnake, Walkinwat, YZHC, zHash, Zitmo, and Zsone.

Therefore, the final dataset is made of a total of 49 samples, one for each family of malware, defined by a total of 26 binary features divided in 6 categories (Table 5.1). A detailed description of each feature can be found in the original paper [6], and some previous work where this dataset is used can be found in [34, 35, 36].

| | |
|---|---|
| **Category 1: Installation** | 1.Repackaging, 2.Update, 3.Drive-by download, 4.Standalone |
| **Category 2: Activation** | 5.Boot, 6.SMS, 7.Net, 8.Call, 9.USB, 10.PKG, 11.Batt, 12.SYS, 13.Main |
| **Category 3: Privilege escalation** | 14.exploid, 15.RATC/zimperlich, 16.ginger break, 17.asroot, 18.encrypted |
| **Category 4: Remote control** | 19.NET, 20.SMS |
| **Category 5: Financial charges** | 21.phone call, 22.SMS, 23.block SMS |
| **Category 6: Personal information stealing** | 24.SMS, 25.phone number, 26.user account |

Tabla 5.1: Features in the Malgenome Dataset

## 5.3. Experiments and Results

This section presents the experiments performed and the results obtained in the validation process of the proposed solution.

Both BHL (section 5.2.1) and DT (section 5.2.2) algorithms have been applied to the previously described dataset (section 5.2.3), in order to identify the features that define the internal structure of the data and that support the grouping of the

different families of malware attacks. In the conducted experiments, firstly BHL is applied to identify groups of malware families with similar behaviour. This is done by visually inspecting the obtained BHL projections, and the most relevant features are consequently identified. Then, the dataset is analyzed by means of DTs to determine the level of importance of each feature, considering as the most relevant features those that are used in the decision nodes at lowest depth.

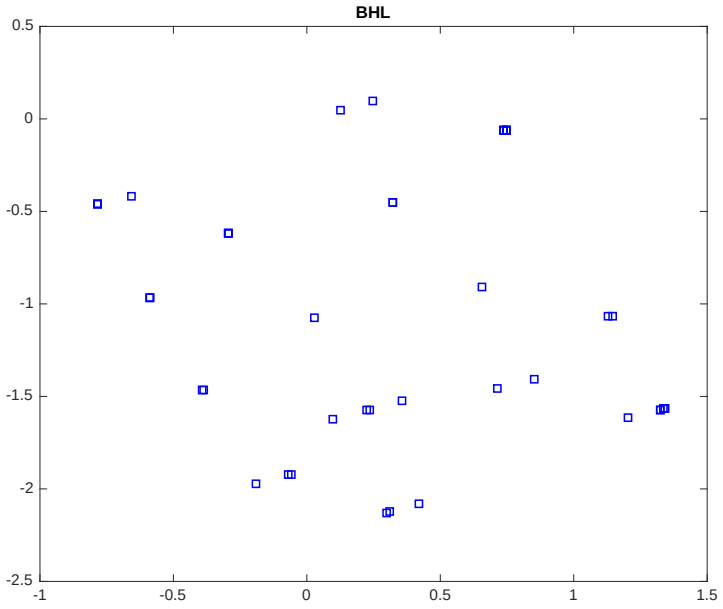


Figura 5.2: BHL - Projection of malware falmilies

In figure 5.2 it is shown the best projection obtained by BHL using the following parameter values: $\alpha = 3$, $\beta = 4$, $number \quad of \quad iterations = 1000$, and $learning \quad rate = 0.05$ . These parameter values were chosen in an experimental process of trial and error. As parameter tuning is a task that is very dependent on the dataset to be used, several initial experiments were conducted with a range of combinations of these parameter values.

Based on such projection, samples are grouped in 2 main clusters: G1 and G2 (figure 5.3). Additionally, several subgroups (at a 3 level depth, ie. G1 $\rightarrow$ G1A $\rightarrow$ G1A.1, G1A.2, G1A.3 and G1A.4) can be defined in these main groups.

Figure 5.4 presents the split of family groups in a schema that shows the results of thoroughly analyzing the allocation of families in groups. It can be seen the most relevant features that have been identified, varying from one cluster to another. As

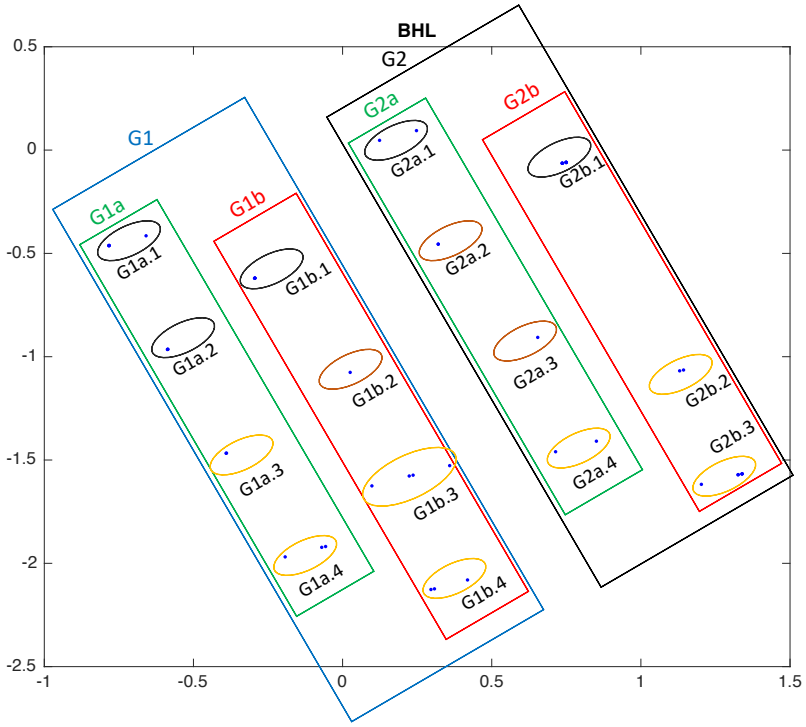


Figura 5.3: BHL - Labelling of clusters

an example, data are split in G1 and G2 based on the features “Repackaging” and “Standalone”. The complete lists of families assigned to each one of the groups are presented in figures 5.5 and 5.6. Malware families are allocated in the same group as they are associated to similar characteristics and behaviour, and therefore there could be similar ways to deal with them.

Based on the analysis of BHL results, the most relevant features, in decreasing order of importance are: “Repackaging” and “Standalone”, “Boot” and “Activation: SMS”, and “Financial Charges: SMS”.

BHL clearly outperforms other algorithms used in previous works [24, 29], providing a clearer visualization of the internal structure of the dataset. Groups obtained by BHL are more compact and well defined than the groups generated by other EPP techniques in the previous work.

In addition to the BHL experiments, experiments with DTs were additionally conducted in order to compare and validate the obtained results. As it has been previously

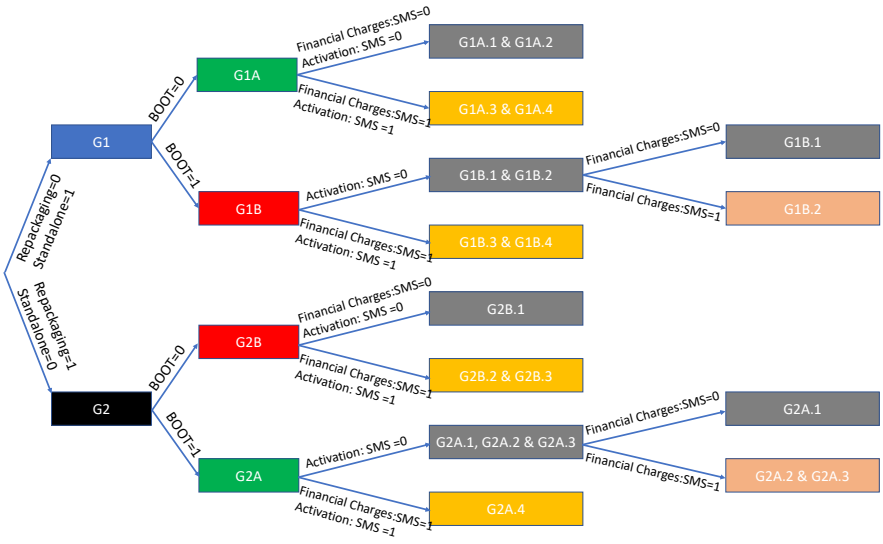


Figura 5.4: Schematic clustering and relevant features from BHL projection

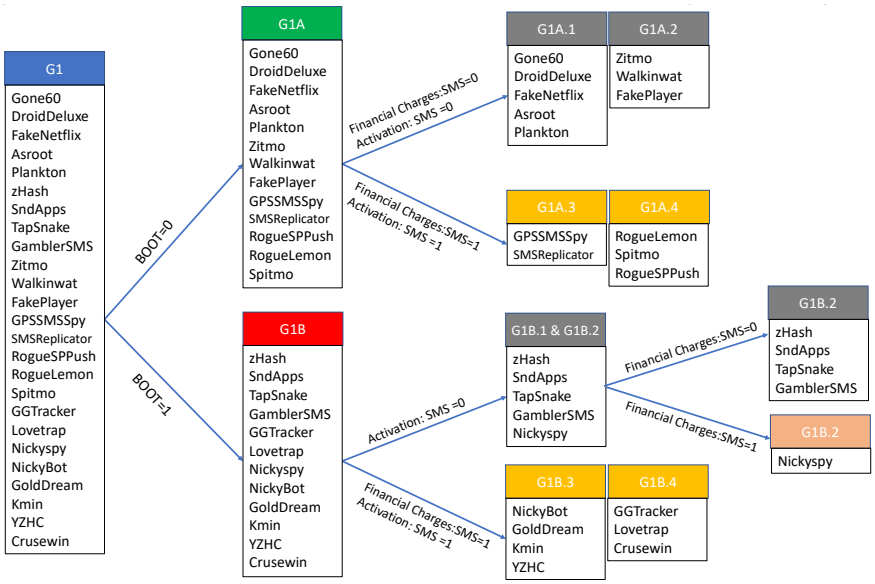


Figura 5.5: Families allocation in Group 1 and relevant features identified in BHL projection

mentioned, 3 different splitting functions have been applied in present paper: Gini, Deviance, and Twoing. In addition, 3 different criteria for selecting split features have been applied: Standard, Curvature, and Interaction.

As an example, one of the obtained DTs is shown in figure 5.7. This is the tree

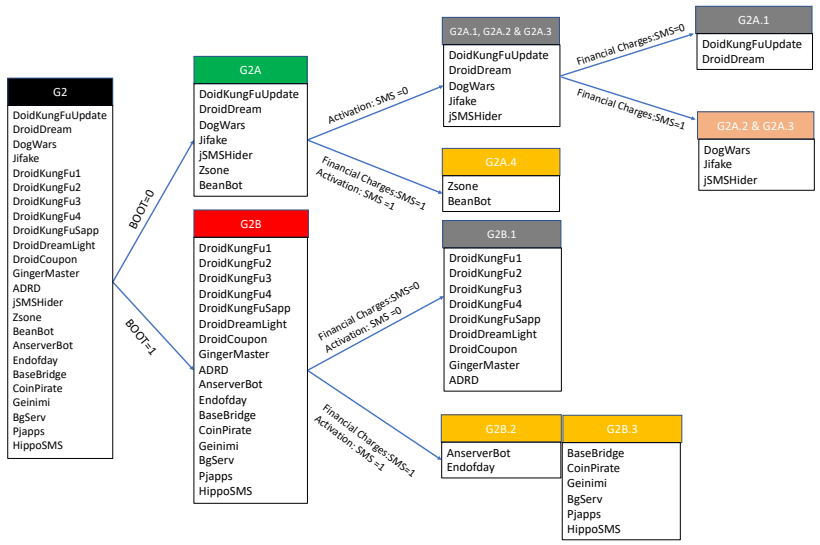


Figura 5.6: Families allocation in Group 2 and relevant features identified in BHL projection

generated from the Malgenome dataset when applying the standard CART split criteria and the Deviance function. It has been selected as it is the one with lowest depth. In the leaf nodes, labels refer to family numbers (alphabetically ordered as presented in Section 5.2.3).

To show the most interesting results from the different alternatives to build DT, information has been summarized in Table 5.2. For each combination of splitting function and selecting criteria, the minimum depth of decision nodes linked to each one of the original features, is shown. That is, when the same feature appears in more than one node, the minimum depth of all these nodes is the one selected for the given feature. In the case a certain feature was not included in the DT, there is no value.

In this table it can be seen that results (slightly or significantly) vary when comparing the obtained results (by different splitting function and selecting criteria) for a certain feature. As general conclusions can not be derived and to sum up all figures, the average depth value is calculated for each feature, that is further analyzed.

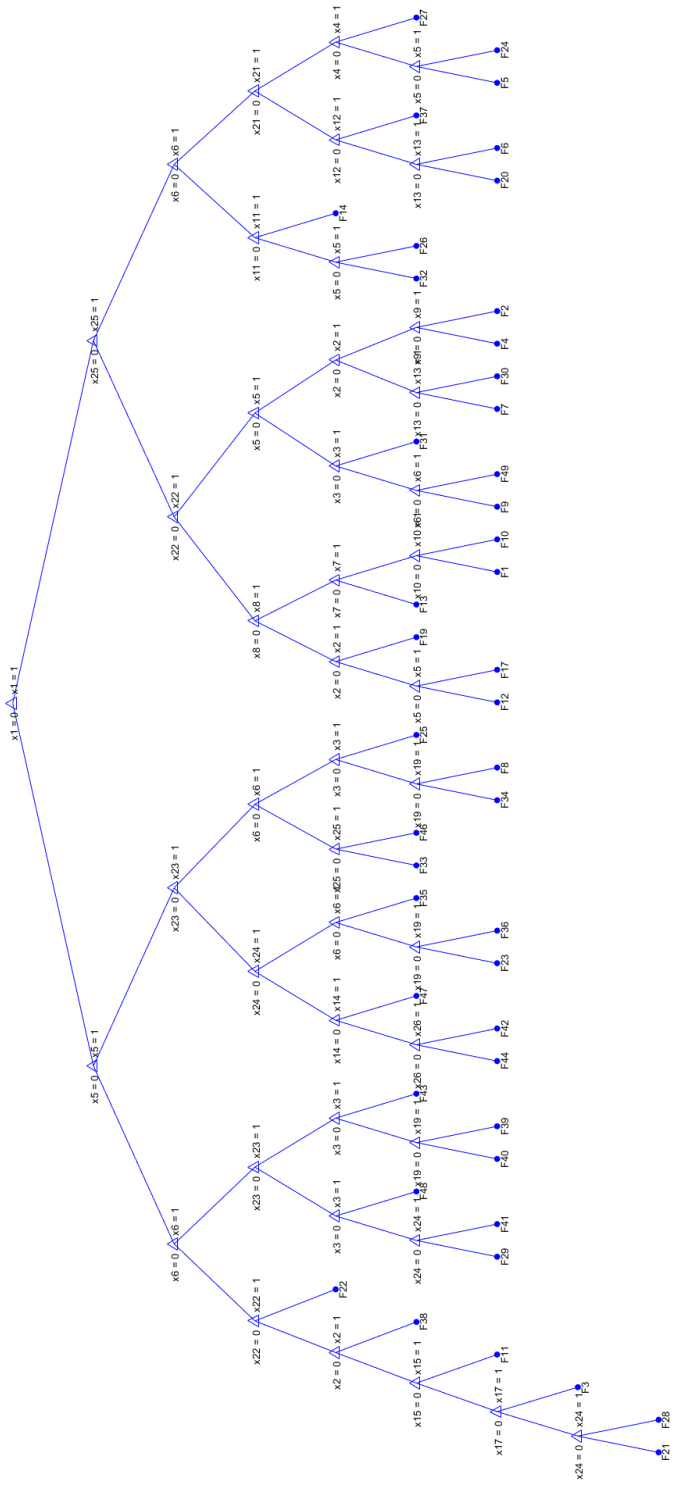

Figura 5.7: DT obtained with standard CART split criteria and Deviance function

| | | Deviance | | | Gini | | | Twoing | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | Feature | Standard | Curvature | Interaction curvature | Standard | Curvature | Interaction curvature | Standard | Curvature | Interaction curvature | Average |
| 1 | Repackaging | 1 | 2 | 4 | 1 | 2 | 6 | 1 | 2 | 4 | 2.56 |
| 5 | BOOT | 2 | 3 | 3 | 4 | 3 | 2 | 2 | 3 | 3 | 2.78 |
| 18 | Encrypted | | 4 | 2 | | 4 | | | 4 | 2 | 3.20 |
| 9 | USB | 6 | 3 | | 5 | 3 | 4 | 6 | 3 | | 4.29 |
| 3 | Drive-by Download | 5 | 5 | 4 | 2 | 5 | 6 | 5 | 5 | 4 | 4.56 |
| 24 | SMS | 4 | 3 | 5 | 10 | 3 | 6 | 3 | 3 | 5 | 4.67 |
| 26 | User Account | 6 | 1 | 8 | 3 | 1 | 8 | 6 | 1 | 8 | 4.67 |
| 2 | Update | 5 | 7 | 4 | 2 | 6 | 3 | 5 | 7 | 4 | 4.78 |
| 19 | NET | 6 | 2 | 8 | 9 | 2 | 3 | 4 | 2 | 8 | 4.89 |
| 6 | SMS | 3 | 6 | 5 | 8 | 7 | 3 | 3 | 6 | 4 | 5.00 |
| 10 | PKG | 6 | 5 | | 4 | 5 | 4 | 6 | 5 | | 5.00 |
| 22 | SMS | 3 | 6 | 4 | 10 | 6 | 4 | 3 | 6 | 4 | 5.11 |
| 4 | Standalone | 5 | 10 | 3 | 3 | 9 | 3 | 5 | 10 | 3 | 5.67 |
| 8 | CALL | 4 | | 7 | 4 | | 8 | 4 | | 7 | 5.67 |
| 11 | BATT | 4 | 9 | | 5 | 4 | 5 | 4 | 9 | | 5.71 |
| 16 | Ginger Break | | | | 6 | | | | | | 6.00 |
| 15 | RATC/Zimperlich | 6 | 8 | 1 | 9 | 9 | 8 | 7 | 8 | 1 | 6.33 |
| 7 | NET | 5 | 8 | 6 | 10 | 9 | 2 | 5 | 8 | 6 | 6.56 |
| 14 | Exploid | 5 | 8 | 6 | 9 | 6 | 6 | 5 | 8 | 6 | 6.56 |
| 17 | Asroot | 7 | 4 | 7 | 11 | 4 | 9 | 8 | 4 | 7 | 6.78 |
| 23 | Block SMS | 3 | 8 | 5 | 9 | 9 | 9 | 4 | 8 | 6 | 6.78 |
| 25 | Phone Number | 2 | 11 | 2 | 12 | 12 | 11 | 2 | 11 | 2 | 7.22 |
| 12 | SYS | 5 | 10 | 6 | 10 | 11 | 7 | 5 | 10 | 6 | 7.78 |
| 21 | Phone Call | 4 | 9 | | 12 | 13 | 7 | 4 | 9 | 5 | 7.88 |
| 13 | MAIN | 6 | 11 | 7 | 10 | 12 | 1 | 6 | 11 | 7 | 7.89 |
| 20 | SMS | | | | 10 | | 8 | | | | 9.00 |

Tabla 5.2: Summary table of DT results: minimum depth of decision nodes for each one of the original features

When analyzing figure 5.4 and table 5.2, it can be seen that results from BHL and DT are coherent. In the case of BHL, it can be seen that Repackaging is identified as the most discriminative feature, because the two main groups in the dataset (G1 and G2) take complementary values for such feature. Coherently, Repackaging is the feature with the lowest mean depth, being included in all the generated trees. Furthermore, it was selected for the root node of 3 DTs. When analyzing subgroups in BHL projection (figure 5.3), it can be seen that BOOT is the feature that drive the split in 1st-level subgroups (subgroups G1A and G1B in the case of group G1, and subgroups G2A and G2B in the case of group G2). In keeping with this idea, according to DTs results, Boot is the second feature with the lowest mean depth. For the next level of importance, the BHL projection identifies Financial Charges SMS and Activation SMS as the features that best explain the split in different subgroups. The two of them are also selected by DTs as ranked in the first half of features with a lowest mean depth, although some other features take lower values.

Additionally, from the DTs results (table 5.2), Privilege escalation - Ginger Break and Remote control - SMS can be identified as the least relevant features. The former was not included in 7 (out of 9) DTs while the latter was not included in 6. Furthermore, Remote control - SMS is the feature with a highest value of the average depth, taking a value of 9. It means that these features are almost useless when characterizing malware families.

Results from present paper are consistent with those obtained in previous work [30] when applying Feature Selection (FS) to the same dataset. Installation - Repackaging, Activation - SMS, Activation - Boot, Remote Control - NET and Financial Charges - SMS were identified as the 5 most relevant features in order to characterize malware families, according to a given method of filter-based FS: Minimum-Redundancy Maximum Relevance. This method is intended at obtaining the maximum relevance to the output while keeping redundancy of selected features to lowest levels. Complementary, two evolutionary approaches to FS (GA-ICC-W and GA-I-W) identified Installation - Repackaging, Installation - Standalone, Activation - SMS, Remote Control - NET, and Financial Charges - SMS as the 5 most relevant ones. These methods perform the selection of features according to the Information Correlation Coefficient and the Mutual Information, respectively.

## 5.4. Conclusions and Future Work

In this paper, some machine learning techniques have been applied to Android malware data in order to analyse the features of such apps and subsequently identify the ones that better define the organization of malware families. As a result, detection and categorization of malware could be improved and sped up at the same time. Furthermore, by knowing about these features, malware apps could be identified more quickly and precisely and then removed from the official Android market.

From the obtained results some conclusions can be derived; first of all, the proposed machine-learning techniques probed to successfully addressed the given challenge. BHL has outperformed previous neural projection techniques that have been applied to the same data in clearly revealing the structure of the Malgenome dataset. Additionally, features identified as the most important ones by such EPP technique are also highlighted by DTs as being relevant to better differentiate between malware families.

Obtained results are consistent with those obtained by FS and hence validate present proposal. Future work will focus on the development of a Hybrid Intelligent System to integrate results from the previously validated machine-learning techniques. In addition, it will be applied to up-to-date malware datasets in order to check its performance when facing 0-day malware.

## 5.5. Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## 5.6. Data Availability

Dataset used in this research is available in [7].

## 5.7. Acknowledgments

This work is partially supported by:

Instituto Nacional de Ciberseguridad (INCIBE) and developed Research Institute of Applied Sciences in Cybersecurity (RIASC).

## Bibliografía

[1] Gartner. (2018) Global smartphone sales to end users from 1st quarter 2009. [Online]. Available: https://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/

[2] AppBrain. (2018) Android and google play statistics. [Online]. Available: https://www.appbrain.com/stats/stats-index

[3] SOPHOSLABS, "Ltd., s., sophoslabs 2019 threat report," Tech. Rep., 2019.

[4] M. LABS, "Labs, m., cybercrime tactics and techniques: Q3 2018," Tech. Rep., 2018.

[5] S. Arshad, M. A. Shah, A. Khan, and M. Ahmed, "Android malware detection & protection: A survey," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, 2016. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2016.070262

[6] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2012, pp. 95–109. [Online]. Available: https://doi.org/10.1109/SP.2012.16

[7] Y. Zhou. (2010) Malgenome project. [Online]. Available: http://www.malgenomeproject.org

[8] H. Quintián and E. Corchado, "Beta hebbian learning as a new method for exploratory projection pursuit," *International Journal of Neural Systems*, vol. 27, no. 6, pp. 1–16, 2017. [Online]. Available: https://doi.org/10.1142/S0129065717500241

[9] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[10] P. G. Nieto, J. M. Torres, F. de Cos Juez, and F. S. Lasheras, "Using multivariate adaptive regression splines and multilayer perceptron networks to evaluate paper manufactured using eucalyptus globulus," *Applied Mathematics and Computation*, vol. 219, no. 2, pp. 755 – 763, 2012. [Online]. Available: https://doi.org/10.1016/j.amc.2012.07.001

[11] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," *Expert Systems with Applications*, vol. 36, no. 1, pp. 2 – 17, 2009. [Online]. Available: https://doi.org/10.1016/j.eswa.2007.10.005

[12] R. Ferreiro García, J. L. Calvo Rolle, M. Romero Gómez, and A. De Miguel Catoira, "Expert condition monitoring on hydrostatic self-levitating bearings," *Expert Systems with Applications*, vol. 40, no. 8, pp. 2975 – 2984, 2013. [Online]. Available: https://doi.org/10.1016/j.eswa.2012.12.013

[13] C. Crespo Turrado, M. d. C. Meizoso López, F. Sánchez Lasheras, B. A. Rodríguez Gómez, J. L. Calvo Rolle, and F. J. De Cos Juez, "Missing data imputation of solar radiation data under different atmospheric conditions," *Sensors*, vol. 14, no. 11, pp. 20 382–20 399, 2014. [Online]. Available: https://doi.org/10.3390/s141120382

[14] J. L. Calvo Rolle, I. Machón González, and H. López García, "Neuro-robust controller for non-linear systems," *Dyna*, vol. 86, no. 3, pp. 308–317, 2011. [Online]. Available: http://dx.doi.org/10.6036/3949

[15] Á. Herrero, E. Corchado, M. A. Pellicer, and A. Abraham, "Hybrid multi agent-neural network intrusion detection with mobile visualization," in *Innovations in Hybrid Intelligent Systems*, 2008, pp. 320–328. [Online]. Available: https://doi.org/10.1007/978-3-540-74972-1_42

[16] R. Sánchez, Á. Herrero, and E. Corchado, "Visualization and clustering for SNMP intrusion detection," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 505–532, 2013. [Online]. Available: https://doi.org/10.1080/01969722.2013.803903

[17] C. Pinzón, Á. Herrero, J. F. de Paz, E. Corchado, and J. Bajo, "Cbrid4sql: A CBR intrusion detector for SQL injection attacks," in *Proceedings of the 5th International Conference on Hybrid Artificial Intelligence Systems HAIS 2010 - Part II*, 2010, pp. 510–519. [Online]. Available: https://doi.org/10.1007/978-3-642-13803-4_63

[18] C. Pinzón, J. F. de Paz, J. Bajo, Á. Herrero, and E. Corchado, "Aiida-sql: An adaptive intelligent intrusion detector agent for detecting sql

injection attacks," in *Proceedings of the 10th International Conference on Hybrid Intelligent Systems HIS 2010*, 2010, pp. 73–78. [Online]. Available: https://doi.org/10.1109/HIS.2010.5600026

[19] D. Atienza, Á. Herrero, and E. Corchado, "Neural analysis of HTTP traffic for web attack detection," in *Proceedings of the 8th International Conference on Computational Intelligence in Security for Information Systems CISIS 2015*, 2015, pp. 201–212. [Online]. Available: https://doi.org/10.1007/978-3-319-19713-5_18

[20] L. Cen, C. S. Gates, L. Si, and N. Li, "A probabilistic discriminative model for android malware detection with decompiled source code," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 400–412, 2015. [Online]. Available: https://doi.org/10.1109/TDSC.2014.2355839

[21] H.-J. Zhu, Z.-H. You, Z.-X. Zhu, W.-L. Shi, X. Chen, and L. Cheng, "Droiddet: Effective and robust detection of android malware using static analysis along with rotation forest model," *Neurocomputing*, vol. 272, pp. 638 – 646, 2018. [Online]. Available: https://doi.org/10.1016/j.neucom.2017.07.030

[22] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," *Soft Computing*, vol. 20, no. 1, pp. 343–357, 2016. [Online]. Available: https://doi.org/10.1007/s00500-014-1511-6

[23] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner, "A Survey of Visualization Systems for Malware Analysis," in *Eurographics Conference on Visualization (EuroVis) - STARs*, 2015. [Online]. Available: https://doi.org/10.2312/eurovisstar.20151114

[24] A. González, Á. Herrero, and E. Corchado, "Neural visualization of android malware families," in *Proceedings of the International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*, 2016, pp. 574–583. [Online]. Available: https://doi.org/10.1007/978-3-319-47364-2_56

[25] A. Paturi, M. Cherukuri, J. Donahue, and S. Mukkamala, "Mobile malware visual analytics and similarities of attack toolkits (malware gene analysis)," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 149–154. [Online]. Available: https://doi.org/10.1109/CTS.2013.6567221

[26] W. Park, K. Lee, K. Cho, and W. Ryu, "Analyzing and detecting method of android malware via disassembling and visualization," in *2014 International Conference on Information and Communication Technology Convergence (ICTC)*,

2014, pp. 817–818. [Online]. Available: https://doi.org/10.1109/ICTC.2014.6983300

[27] V. Moonsamy, J. Rong, and S. Liu, "Mining permission patterns for contrasting clean and malicious android applications," *Future Generation Computer Systems*, vol. 36, pp. 122 – 132, 2014. [Online]. Available: https://doi.org/10.1016/j.future.2013.09.014

[28] O. Somarriba, U. Zurutuza, R. Uribeetxeberria, L. Delosieres, and S. Nadjm-Tehrani, "Detection and visualization of android malware behavior," *Journal of Electrical and Computer Engineering*, vol. 2016, 2016. [Online]. Available: http://dx.doi.org/10.1155/2016/8034967

[29] R. Vega Vega, H. Quintián, J. L. Calvo-Rolle, Á. Herrero, and E. Corchado, "Gaining deep knowledge of android malware families through dimensionality reduction techniques," *Logic Journal of the IGPL, Ïn press"*, 2019. [Online]. Available: http://dx.doi.org/10.1093/jigpal/jzy030

[30] J. Sedano, S. González, C. Chira, Á. Herrero, E. Corchado, and J. R. Villar, "Key features for the characterization of android malware families," *Logic Journal of the IGPL*, vol. 25, no. 1, pp. 54–66, 2017. [Online]. Available: https://doi.org/10.1093/jigpal/jzw046

[31] S. Singh and P. Gupta, "Comparative study id3, cart and c4. 5 decision tree algorithm: a survey," *International Journal of Advanced Information Science and Technology*, vol. 27, no. 7, pp. 97–103, 2014.

[32] W.-Y. Loh and Y.-S. Shih, "Split selection methods for classification trees," *Statistica sinica*, vol. 7, no. 4, pp. 815–840, 1997.

[33] W.-Y. Loh, "Regression trees with unbiased variable selection and interaction detection," *Statistica Sinica*, vol. 12, no. 12, pp. 361–386, 2002.

[34] L. Li, A. Bartel, T. F. BissyandÃ©, J. Klein, Y. L. Traon, S. Arzt, S. Rasthofer, E. Bodden, D. Octeau, and P. McDaniel, "Iccta: Detecting inter-component privacy leaks in android apps," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 1, 2015, pp. 280–291. [Online]. Available: https://doi.org/10.1109/ICSE.2015.48

[35] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket," in *2014 Network and Distributed System Security (NDSS) Symposium*, vol. 14, 2014, pp. 23–26.

[36] G. Suarez-Tangil, S. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "Droidsieve: Fast and accurate classification of obfuscated android malware," in *Seventh ACM on Conference on Data and Application Security and Privacy (CODASPY)*, 2017. [Online]. Available: https://doi.org/10.1145/3029806.3029825

# CAPÍTULO 6

# Intrusion Detection with Unsupervised Techniques for Network Management Protocols over Smart Grids

En este capítulo se presenta el tercero de los artículos utilizado para la defensa de la tesis por compendio de artículos. A continuación se presentan los datos básicos del mismo, y después de incluirá todo el contenido del artículo copiado del artículo publicado. Se ha decidido no incluir el artículo original para mantener una uniformidad en la presentación de la tesis, pero se puede consultar a través del *doi* correspondiente.

Datos generales del artículo:

**Autores:**
Rafael Alejandro Vega Vega[1], Pablo Chamoso-Santos[2,3], Alfonso González Briones[2,3,4], José-Luis Casteleiro-Roca[1], Esteban Jove[1], María del Carmen Meizoso-López[1], Benigno Antonio Rodríguez-Gómez[1], Héctor Quintián[1], Álvaro Herrero[5], Kenji Matsui[6]and Emilio Corchado[2] and José Luis Calvo-Rolle[1]

**Afiliaciones:**
[1] Department of Industrial Engineering, University of A Coruña, 15403 Ferrol, Spain rafael.alejandro.vega.vega@udc.es, jose.luis.casteleiro@udc.es, esteban.jove@udc.es, carmen.meizoso@udc.es, benigno.rodriguez@udc.es, hector.quintian@udc.es, jlcalvo@udc.es
[2] BISITE Research Group, University of Salamanca, Edificio I+D+i, Calle Espejo 2, 37007 Salamanca, Spain; chamoso@usal.es, alfonsogb@usal.es, escorchado@usal.es
[3] Air Institute, IoT Digital Innovation Hub (Spain), Calle Segunda 4, 37188 Salamanca, Spain
[4] Research Group on Agent-Based, Social and Interdisciplinary Applications (GRASIA), Complutense University of Madrid, 28040 Madrid, Spain
[5] Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos Spain; ahcosio@ubu.es
[6] Faculty of Robotics & Design, Osaka Institute of Technology, Osaka 535-8585, Japan; kenji.matsui@oit.ac.jp

**Título:** Intrusion Detection with Unsupervised Techniques for Network Management Protocols over Smart Grids
**Revista:** Applied Sciences
**Publicación:** Vol.10, no.7
**Páginas:** 1–13
**Editorial:** MDPI
**Año de publicación:** 2020
**DOI:** 10.3390/app10072276
**Factor de impacto JCR 2020:** 2,679
**Factor de impacto JCR 2020 (5 años):** 2,736
**Disciplina *Physics, Applied (SCIE)* 2020:** 73/160 - Q2
**Disciplina *Engineering, Multidisciplinary (SCIE)* 2020:** 38/90 - Q2
**Disciplina *Materials Science, Multidisciplinary (SCIE)* 2020:** 201/334 - Q3
**Disciplina *Chemistry, Multidisciplinary (SCIE)* 2020:** 101/178 - Q3

## Abstract

The present research work focuses on overcoming cybersecurity problems in the Smart Grid. Smart Grids must have feasible data capture and communications infrastructure to be able to manage the huge amounts of data coming from sensors. To ensure the proper operation of next-generation electricity grids, the captured data must be reliable and protected against vulnerabilities and possible attacks. The contribution of this paper to the state of the art lies in the identification of cyberattacks that produce anomalous behaviour in network management protocols. A novel neural projectionist technique (Beta Hebbian Learning, BHL) has been employed to get a general visual representation of the traffic of a network, making it possible to identify any abnormal behaviours and patterns, indicative of a cyberattack. This novel approach has been validated on 3 different datasets, demonstrating the ability of BHL to detect different types of attacks, more effectively than other state-of-the-art methods.

**keywords:** smart grid; computational intelligence; automatic response; exploratory projection pursuit; neural networks

## 6.1. Introduction

Care for the environment is not a simple trend. It is a very important matter from a legal point of view. Governments have already implemented regulations, making it compulsory to take action against environmental degradation, and there will certainly be more regulations in the future. It is necessary to remark that zero impact is impossible from a practical point of view. Nevertheless, it is necessary to pursue sustainability and to minimize impact [1]. Renewable energy systems play a very important role [2]. The environmental impact caused by this type of systems is much lesser than of conventional sources, especially when their useful life is taken into account [3].

From a theoretical point of view, depletable resources should be fully replaced by renewable energy. However, if we consider the electric sector as a global unit, our current possibilities are still too limited. In fact, several state-of-the-art studies have concluded that increasing the use of renewable energies could destabilize the energy system [4, 5].

Some highly developed countries have implemented regulations that make the use of

renewable energy sources obligatory, especially in new buildings. However, the connection of those buildings to the power network makes energy management very difficult. This is because, even when they generate energy that is not electricity, they can still cause the energy demand to reduce.

The main problem of the electric sector is that the levels of energy production must be equivalent to the amount of energy being consumed [6]. This justifies the need for energy storage systems which mitigate problems associated with unbalanced generation and consumption levels [7].

Thanks to this kind of system, when excess energy is generated, the excess consumption can be stored, similarly, when the energy needs are greater than the amount of generated energy, the storage system may supply the required energy. The main problem currently is that energy storage systems are inefficient [8].

Considering the problems described above, the optimal management of every part of the power network is mandatory. However, to make efficient management possible, it is necessary to develop adequate tools that will ensure the correct performance of the system as a whole [9]. The term Smart Grid [9, 10] emerged as an answer to all the issues described above. The Smart Grid makes it possible to measure the levels of energy generation/consumption and forecast the future levels of both variables, making it possible to manage the entire system more effectively. Nevertheless, the task of precisely adjusting energy generation to demand continues to be very complex, thus, it is preferable to use an energy storage system [7].

From a global context, a smart grid can be defined as the dynamic integration of developments in electrical engineering and energy storage, the advances in information and communication technologies (or ICT), their implementation in the electricity-related processes (generation, transport, distribution, storage and marketing, including alternative energy) [11]. ICT makes it possible to concatenate security, control, instrumentation, measurement, quality and administration of energy, etc., in a single management system, with the primary objective of making efficient and rational use of electricity [12].

The concept described above could also include the integration of other actors in the area of measurement and control, such as gas sources and water services. Thus, smart electricity networks become part of a macro-concept of territorial dominance, such as that of smart cities [13]. The smart grid is a type of efficient electricity management that

uses computer technology to optimize the production and distribution of electricity, in order to achieve a greater balance between supply and demand, as well as producers and consumers [11].

The smart grid must be protected from all types of vulnerabilities, like natural disasters, and of course, it must be robust against attacks [14]. Security is essential because otherwise the information flow between all the actors would not be reliable [15]. In consequence, the smart grid concept would fail completely [16]. Robust communications and the reliability of the information must be guaranteed in all cases for satisfactory smart grid performance [12, 14].

Among the actors in the Smart Grid, there are three crucial components to which special attention should be paid [9]: data acquisition, data management and communications. It is necessary to ensure secure communications and the reliability of the available data. Moreover, protection mechanisms must be implemented for protection against any type of attack. The above-mentioned goals may be achieved thanks to advances in cybersecurity [17].

Cybersecurity has become a relevant field in multiple areas and it is the basis of the proposed solution.

The main objective of this research is to identify cyberattacks which produce anomalous behaviours in network management protocols. This has been made possible through the use of a novel neural projectionist technique called Beta Hebbian Learning (BHL), which provides a visual representation of the network traffic and detects abnormal network behaviours and patterns, indicative of a cyberattack.

The rest of the paper is structured as follows: Section 6.2 presents a review of sate-of-the-art research in the field. Section 6.3 describes the main materials and methods used in this research, including the datasets, and the novel Beta Hebbian Learning algorithm used for attack detection. The next section details the results of each of the experiments performed on the real datasets, and finally, Section 6.5 presents the conclusions.

## 6.2. Literature Review

This section presents related state-of-the-art literature and the principal advantages of the proposed model.

Several authors carried out research on building a system for real-time intrusion detection by training it with a dataset. However, current systems are only able to detect some but not all the indications of an intrusion. This is because they are not able to monitor all the behaviours in the network On the contrary, projectionist techniques are able to provide a visual overview of the network traffic. Earlier dimensionality reduction techniques were applied to visualize network data using scatter plots [18, 19, 20, 21, 22, 23]. In the case of [24], several projectionist algorithms, such as PCA, CMLHL, CCA, and SOM network, have been applied to monitor the traffic of the Euskalert network (Honeynet data) [25], to discover behaviour and strategies indicative of an attack. In [26], the same techniques have been applied to GICAP-IDS and DARPA datasets [27], and their performance has been measured according to different variables, such as data volume, system dynamics and network traffic diversity, including first-time attacks (0-day). Then, the authors presented a novel Multi-Agent System which combined Artificial Neural Networks (ANN) with Case-Based Reasoning (CBR) techniques for the detection of attacks in computer networks [28]. This new IDS, known as RT-MOVICABIDS, has been validated using three different datasets. Those datasets have also been used in our study, as described further on in the article. In [29], clustering and visualization techniques have been combined to generate an automatic response to the previously developed MOVICAB-IDS system. The modified MOVICAB-IDS has been applied to the three datasets, to assess the improvement of the proposed approach. Furthermore, in [30], it has been validated using a community search dataset. This type of attack involves guessing the password, it has been detected by MOVICAB-IDS, which demonstrated to perform better than other well-know algorithms for detecting attacks on continuous network flow. Finally, in [31], MOVICAB-IDS has been applied to a dataset that contained flow-based information (14.2 M flows). The University of Twente [32] collected this information in September 2008, using a honeypot.

More recently, a novel EPP algorithm, BHL, has been applied to Android malware families [33, 34], obtaining much better results than other well-known algorithms. BHL has also been previously employed in the analysis of the internal structure of a series of datasets [35, 36], providing a clear projection of the original dataset. More specifically,

it has been successfully applied to Android malware datasets [33, 34], where its task was to characterize Android malware families. Therefore, this research aims to apply BHL to the datasets that have previously been used by MOVICAB-IDS, with the aim of improving the obtained projections and achieving a better visual representation of the network traffic. This facilitates the early identification of anomalous situations which may be indicative of a cyberattack in the computer network.

## 6.3. Materials and Methods

In this research, the Exploratory Projection Pursuit (EPP), called Beta Hebbian Learning algorithm (BHL) [37], has been employed. It is based on beta distribution and has been applied to 3 real datasets in order to assess its ability to detect anomalous situations in the network management protocol. Its performance has been compared with the results obtained by the MOVICAB-IDS algorithm [29].

### 6.3.1. Preprocessing

Before using the obtained dataset, they had to undergo a preprocessing stage. First, all missing values were removed.

Outliers have been removed in order to prevent them from being identified as intrusion samples, as this would have affected the training process. Considering as outliers the samples with values outside the $\mu \pm 5\sigma^2$ range, where $\mu$ is the average and $\sigma^2$ is the variance.

The application of this criterion may lead to a situation where some outliers could be considered as intrusion samples. However, their influence on the training process would be insignificant, given that their degree of deviation from the mean would have been small. Although once the system is trained these extreme outliers could be identified as intrusions, a real intrusion is never considered as normal behaviour due to the influence of the outliers during the training process.

Finally a normalization of each variable between the range -1 to 1 has been applied to ensure the stability of the BHL network during the training process [37].

### 6.3.2. Beta Hebbian Learning Algorithm

Artificial Neural Networks (ANN) are typically software simulations that emulate some of the features of real neural networks found in the animal brain. Among the range of applications of unsupervised artificial neural networks, data projection or visualization is the one that facilitates, human experts, the analysis of the internal structure of a dataset. This can be achieved by projecting data on a more informative axis or by generating maps that represent the inner structure of datasets. This kind of data visualization can usually be achieved with techniques such as Exploratory Projection Pursuit (EPP) [37, 38] which project the data onto a low dimensional subspace, enabling the expert to search for structures through visual inspection. The Beta Hebbian Learning technique (BHL) [31] is an Artificial Neural Network belonging to the family of unsupervised EPP, which uses Beta distribution as part of the weight update process, for the extraction of information from high dimensional datasets by projecting the data onto low dimensional (typically 2 dimensional) subspaces. This technique is better than other exploratory methods in that it provides a clear representation of the internal structure of data. The Beta Hebbian Learning network is based on a Negative Feedback Network, therefore to introduce it, consider an N-dimensional input vector, $x$, and a M-dimensional output vector, $y$, where $W_ij$ is the weight linking input $j$ to output $i$ and let $\eta$ be the learning rate. The initial situation is that there is no activation at all in the network. The input data is feedforward via weights from the input neurons (the x-values) to the output neurons (the y-values), where a linear summation is performed to activate the output neuron (see Figure 6.1). This is expressed by Equation (6.1).

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{6.1}$$

The activation is feedback through the same weights and is subtracted from the inputs (see Equation (6.2)).

$$Feedback : e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i \tag{6.2}$$

After that, simple Hebbian learning is performed between input and outputs, the weight update is obtained by means of Equation (6.3).

$$\Delta W_{ij} = \eta e_j y_i \tag{6.3}$$

The effect of the negative feedback is to stabilize the learning in the network. For

this reason, it is not necessary to normalize or clip the weights to achieve a stable solution.

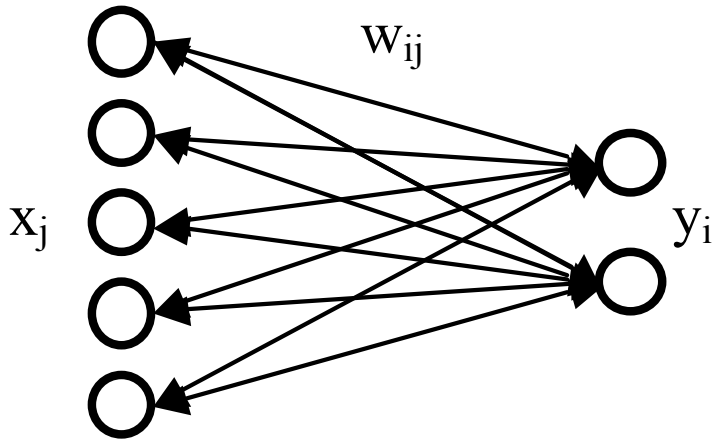


Figura 6.1: Basic architecture of a negative feedback network.

Note that this algorithm is clearly equivalent to Ojâs Subspace Algorithm (Equation (6.4)).

$$\Delta W_{ij} = \eta(x_j - \sum_{i=1}^{M} W_{ij} y_i) y_i \tag{6.4}$$

This network is capable of finding the principal components of the input data in a manner that is equivalent to Oja's Subspace algorithm. Thus, it may be said that the network uses simple Hebbian learning to enable the weights to converge and extract the maximum content from the input data.

Since the model is equivalent to Oja's Subspace algorithm, we might legitimately ask what we gain by using the negative feedback in this way.

Writing the algorithm in this way, gives a model of the process which allows to devise different versions and algorithms like the Beta Hebbian Learning rule. This rule is based on an explicit view of the residual which is never independently calculated using e.g., Oja's learning rule.

A general cost function associated with the Beta Hebbian Learning network can be denoted as Equation (6.5)

$$J = E(-p(e)) \tag{6.5}$$

where $E$ is the expected value operator.

Therefore, the gradient descent J is presented in Equation (6.6).

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial e}\frac{\partial e}{\partial W} \tag{6.6}$$

Thus, the optimal cost function can be obtained if the PDF of the residuals is known. Therefore, the residual ($e$) can be expressed by Equation (6.7) in terms of Beta distribution parameters ($B(\alpha, \beta)$):

$$p(e) = e^{\alpha-1}(1-e)^{\beta-1} = (x - Wy)^{\alpha-1}(1 - x + Wy)^{\beta-1} \tag{6.7}$$

where $\alpha$ and $\beta$ control the PDF shape of the Beta distribution, $e$ is the residual, $x$ are inputs of the network, $W$ is the weight matrix, and $y$ is the output of the network. Finally, gradient descent can be used to maximize the likelihood of the weights (Equation (6.8)):

$$\begin{aligned}
\Delta W \propto \frac{\partial p_i}{\partial W_{ij}} = \frac{\partial}{\partial W_{ij}}[(x_j - W_{ij}yi)^{\alpha-1}(1 - x_j + W_{ij}y_i)^{\beta-1}] = \\
[(\alpha-1)(x_j - W_{ij}yi)^{\alpha-2}(-y_i)(1 - x_j + W_{ij}y_i)^{\beta-1}] + \\
[(x_j - W_{ij}y_i)^{\alpha-1}(\beta-1)(1 - x_j + W_{ij}yi)^{\beta-2}y_i] = \\
[(\alpha-1)e_j^{\alpha-2}(-y_i)(1-e_j)^{\beta-1}] + [e_j^{\alpha-1}(\beta-1)(1-e_j)^{\beta-2}y_i] = \\
y_i e_j^{\alpha-2}[(\alpha-1)(-1)(1-e_j)^{\beta-1} + e_j(\beta-1)(1-e_j)^{\beta-2}] = \\
y_i e_j^{\alpha-2}(1-e_j)^{\beta-2}[(\alpha-1)(-1)(1-e_j) + e_j(\beta-1)] = \\
y_i e_j^{\alpha-2}(1-e_j)^{\beta-2}[(-\alpha + e_j\alpha + 1 - e_j + e_j\beta - e_j)] = \\
y_i e_j^{\alpha-2}(1-e_j)^{\beta-2}[(e_j(\alpha+\beta-2) + 1 - \alpha)]
\end{aligned} \tag{6.8}$$

Therefore, a BHL architecture can be expressed by means the following equations:

$$Feedforward : y_i = \sum_{j=1}^{N} W_{ij}x_j, \forall i \tag{6.9}$$

$$Feedback : e_j = x_j - \sum_{i=1}^{M} W_{ij}y_i \tag{6.10}$$

$$Weightsupdate : \Delta W_{ij} = \eta(e_j^{\alpha-2}(1-e_j)^{\beta-2}(1-\alpha+e_j(\alpha+\beta-2)))y_i \tag{6.11}$$

where $\eta$ is the learning rate.

For the final implementation of the algorithm, the absolute vale of the error where used and finally the sign operator where added to the final result in the weights update.

### 6.3.3. Dataset

In this research, BHL has been applied to 3 real datasets. Each dataset consists of the monitorization of simple networks where different anomalous situations occur. The dataset analyzed in present research has been generated in a small-size university network.

In all cases, the same 5 variables were monitored:

- Packet ID.
- Timestamp: respect to the first captured packet.
- Source Port: It is the host port from which the packet is sent.
- Destination Port: It is the host port to which the packet is sent.
- Packet Size.
- Protocol ID: from 1 to 35 for different packet protocols.

Each dataset contains the data that had been collected during the monitoring of a network in a period where a specific attack occured. The type of attack is different in each dataset. A small part of the data is captured for analysis.Consequently, only the above-mentioned 5 fields of packet headers are used [23], and one output variable is only used to show the real category (normal and attack) but it is never used for training.

## 6.4. Experiments and Results

The BHL algorithm is applied as a clustering technique to identify the internal structure of the 3 datasets and any anomalous situations present in each one. As, the dataset condition in a great manner the selection of optimal parameters, different values combinations of $\alpha$ and $\beta$ parameters were tried and the best combination was selected (Table 6.1). Once the best parameters were obtained, several runs with random weights initialization were performed for validating the repeatability of the obtained results.

Tabla 6.1: Dataset description.

| **Dataset** | **Description** | **Nº Samples** | **Nº of Attacks** |
|---|---|---|---|
| 1 | Type of attack: Scans. In this type of attack, diverse messages are sent to various host ports to extract information about the activity status. An external agent could send these messages with the aim of getting information about host network services. However, in the case of a network scan, the target of several hosts is a specific port (frequently, a single IP address range for all hosts). The target port numbers are 161, 162, and 3750 in the same IP address range for all machines. | 866 | 18 attacks × 3 ports = 54 attacks |
| 2 | Type of attack: MIB (management information base) Information Transfer. In this attack part of the information (or all) of SNMP MIB is captured, usually by means of get/getbulk command, which represents a potentially dangerous situation. However, some queries of MIB could belong to a "normal"· network behavior. | 5000 | 226 attacks |
| 3 | Type of attack: It is a combination of Scan and MIB Information Transfer. | 5866 | 18 attacks × 3 ports = 54 port attacks and 226 MIB attacks |

In the case of the k-means algorithm, to ensure good results in the creation of the cluster, the k-means algorithm was random initialization of the centroids, and the training was repeated 20 times. These repetitions allow avoiding to finish the training in a local minimum.

In all cases, Matlab software was used to analyze the 3 datasets with both algorithms. The implmentation of BHL was done according to previous researches [37] and the Matlab version of the k-means algorithm was used.

Table 6.2 shows the best combination of parameters and Figures 6.1–6.6 the pro-

jections for each dataset and algorithm (BHL and k-means).

Tabla 6.2: BHL and k-means parameters for datasets 1, 2 and 3.

| **Algorithm** | **Parameters** |
|---|---|
| BHL (dataset 1) | iters = 3000, lrate = 0.01, $\alpha = 3$, $\beta = 3$ |
| BHL (dataset 2) | iters = 10,000, lrate = 0.01, $\alpha = 3$, $\beta = 4$ |
| BHL (dataset 3) | iters = 10,000, lrate = 0.05, $\alpha = 3.5$, $\beta = 5$ |
| k-means (dataset 1, 2 and 3) | k = 6, random initialization of the centroids, sqEuclidean distance |

The axes of BHL projections correspond to the first two components of the new subspace (as non-linear combinations of the original space), which do not have any meaning or direct relationship with the variables of the original dataset.

Figure 6.2, shows the best projection of BHL for dataset 1, it shows that BHL can clearly identify 3 clusters which correspond to each scan port attack (port numbers 161, 162, and 3750).

Figure 6.3 shows the results obtained in past researches. It can be observed that, like BHL, there also were 3 clusters in the results, which correspond to different types of scan port attacks. Therefore, due to the simplicity of this dataset, it is not possible to get better results, and in all cases, the scan port attack can be identified easily (k-means clustering was applied to the projected data representation).

Nevertheless, in the case of dataset 2, the are significant differences. In the case of BHL, its projection shows a clear distinction between a normal (green crosses Figure 6.4), and an abnormal situation (MIB transfer source-destination, red dots in Figure 6.5). In the case of MOVICAB-IDS, the final projection mixes both classes (normal and attack), in different clusters (see Figure 6.5).

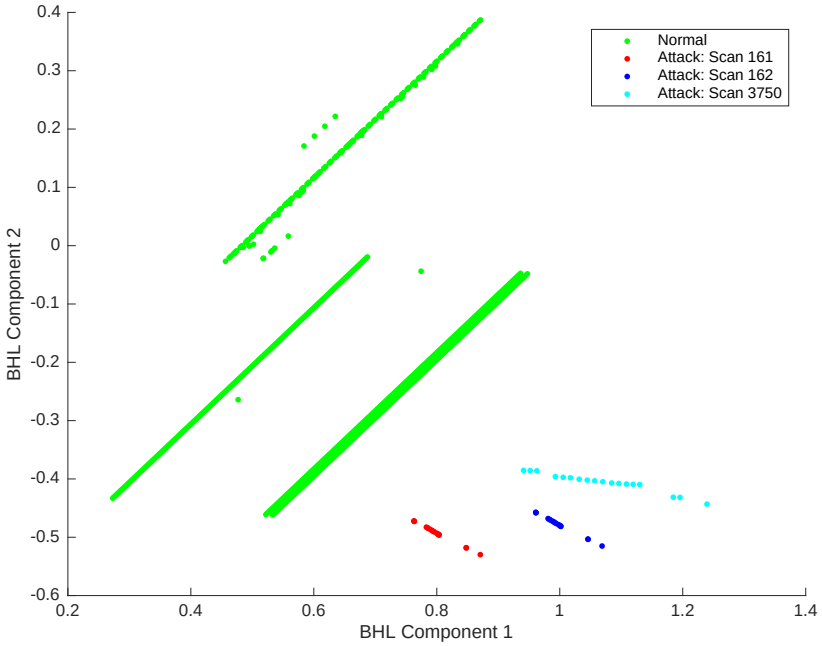


Figura 6.2: BHL projection for dataset 1, port scan attack.

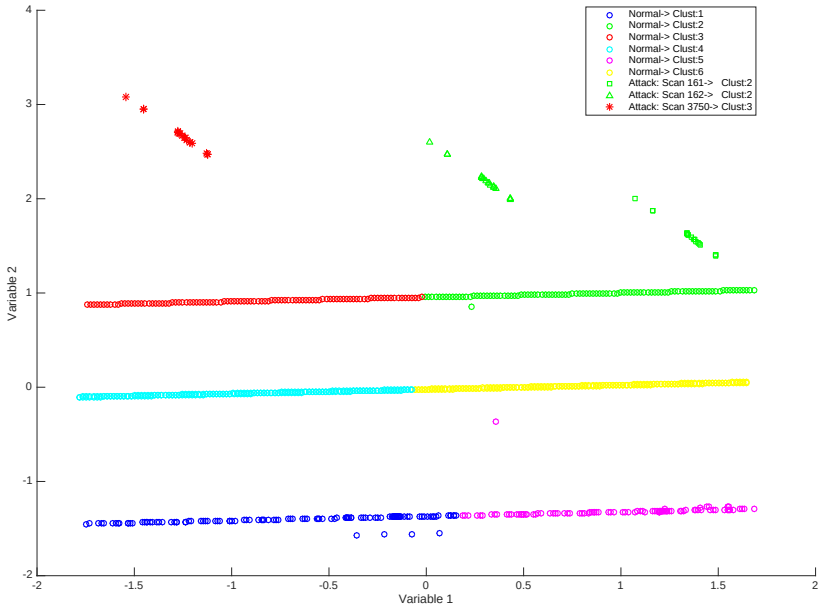


Figura 6.3: MOVICAB projection for dataset 1, port scan attack.

Finally, dataset 3 presents a combination of 2 types of attacks, scan port (3 port attacks) and MIB transfer (source-destination and destination-source attacks), therefore, there are 5 attacks and the rest of the sample contains information about the normal

behaviour of the network. Figure 6.6 presents the best BHL projection for dataset 3 (MIB transfer and scan port). In this case, BHL can clearly differentiate between the samples associated with normal network behaviour (green samples in Figure 6.6) and the samples belonging to abnormal situations. Moreover, BHL can distinguish the different types of scan port (red, blue and cyan samples in Figure 6.6) and MIB transfer attacks (2 types, yellow and magenta samples in Figure 6.6).

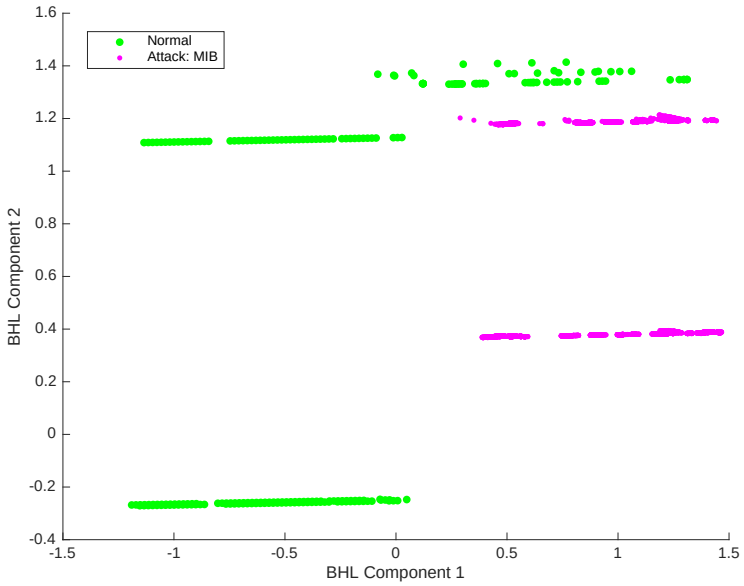


Figura 6.4: BHL projection for dataset 2, MIB transfer attack.

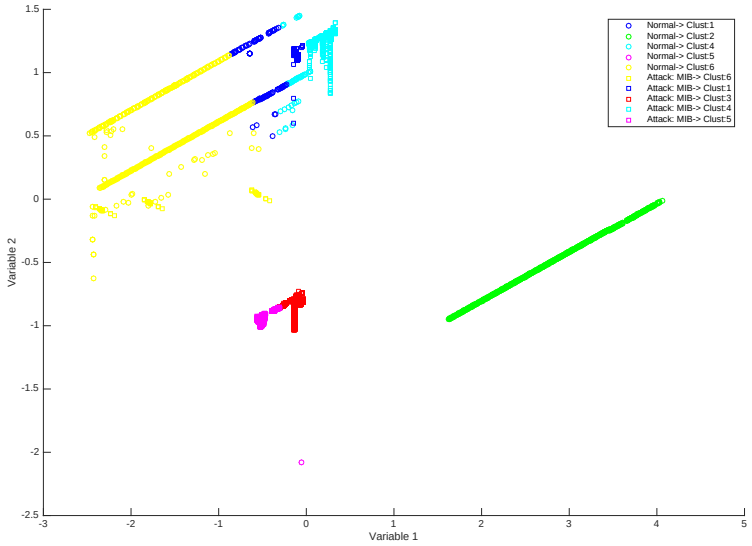


Figura 6.5: MOVICAB projection for dataset 2, MIB transfer attack.

However, in previous researches (see Figure 6.7), MOVICAB-IDS was not able to generate separate groups without mistakes, as it mixed packets belonging to different categories, failing to distinguish between normal and abnormal samples. It is important to remark, that MOVICAB-IDS was able to detect 3 classes; normal samples, scan port, and MIB, however, it was not able to distinguish between the different types of attacks; as can be seen in Figure 6.7, it confused MIB transfer with normal samples. On the contrary, BHL is able to clearly distinguish between all the types of attacks, including destination-source and source-destination MIB transfer attacks.

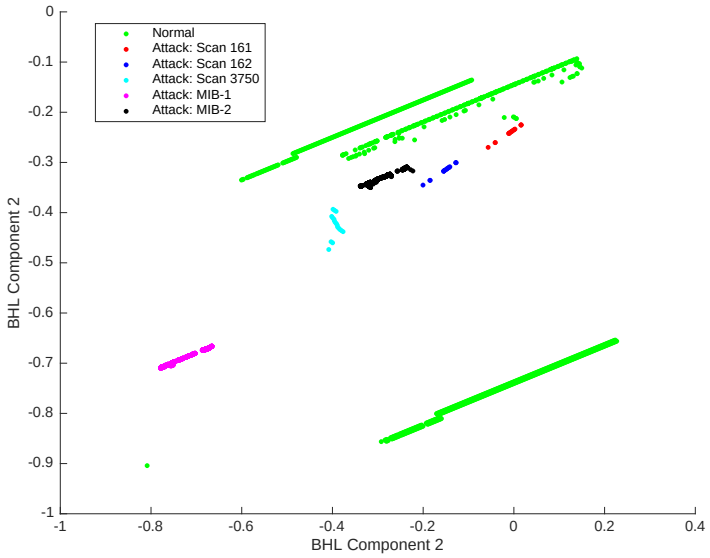


Figura 6.6: BHL projection for dataset 3, MIB transfer and port scan attacks.

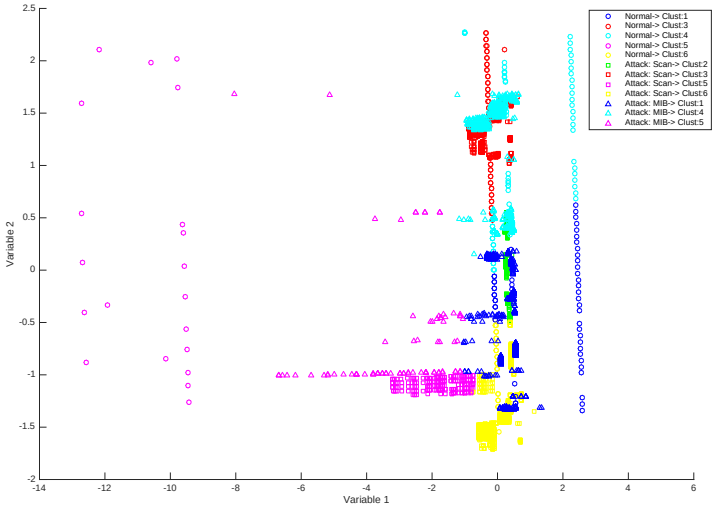


Figura 6.7: MOVICAB projection for dataset 3, MIB transfer and port scan attacks.

## 6.5. Conclusions

The increase in data traffic in smart grids makes them increasingly vulnerable to cyberattacks. Having the tools that permit the correct analysis and visualization of data traffic in these networks becomes increasingly important. Therefore, the use of tools that are capable of visually representing the general behavior of these networks (in terms of data traffic), allows to quickly and easily detect possible attacks that cause abnormal network behavior.

The results presented in Section 6.4 demonstrate that Dimensional Reduction Techniques (DRT), provide a general overview of the internal dataset structure. This helps prevent potential cyberattacks in smart grids through the visual inspection of the network's traffic data.

The previously applied DRTs were able to visually represent the behaviour of the network's traffic data, however, their clustering is not good enough, especially in the case of different types of attacks that are produced at the same time (MIB and Scan port).

On the contrary, BHL gives a detailed overview of the network traffic and provides well-defined clusters that make it possible to identify anomalous situations and different types of attacks, overcoming the challenges associated with data volume, system dynamics and network traffic diversity, including first-time attacks (0-day).

The clarity of BHL projections makes it easy to distinguish between normal traffic and anomalous traffic patterns, facilitating the early detection of attacks. BHL can easily identify scan port attacks at different ports. Moreover, when this type of attack is combined with the MIB attack, BHL is not only able to distinguish between them but also between the two types of MIB attack and the scanned ports. This makes BHL a powerful tool for network management protocols.

The results of the conducted experiment have proven that BHL's performance is superior to that of the techniques used in previous researches, proving comprehensible projections, where attacks are clearly distinguished from the normal behaviour of the network, even when different types of attacks occur at the same time.

The results obtained by EPP algorithms such as BHL demonstrate that they are

suitable to be applied in smart grids for the detection of intrusions in the network. In conclusion, advances have been made in the early identification and characterization of cyberattacks. However, there is still a lot of room for improvement, especially in relation to the security of Smart Grids.

## Bibliografía

[1] T. Kuwae and M. Hori, "Global environmental issues," *Blue Carbon in Shallow Coastal Ecosystems: Carbon Dynamics, Policy, and Implementation*, 2019.

[2] H. Karunathilake, K. Hewage, W. Mérida, and R. Sadiq, "Renewable energy selection for net-zero energy communities: Life cycle based decision making under uncertainty," *Renewable energy*, vol. 130, pp. 558–573, 2019.

[3] R. Prakash, I. K. Bhat *et al.*, "Energy, economics and environmental impacts of renewable energy systems," *Renewable and sustainable energy reviews*, vol. 13, no. 9, pp. 2716–2721, 2009.

[4] S. Chen, F. Zhu, H. Long, and J. Yang, "Energy footprint controlled by urban demands: How much does supply chain complexity contribute?" *Energy*, vol. 183, pp. 561 – 572, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036054421931299X

[5] G. Carrosio and I. Scotti, "The 'patchy' spread of renewables: A socio-territorial perspective on the energy transition process," *Energy Policy*, vol. 129, pp. 684 – 692, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301421519301405

[6] J.-A. Montero-Sousa, J.-L. Casteleiro-Roca, and J.-L. Calvo-Rolle, "Evolution of the electricity sector after the 2nd world war," *DYNA*, vol. 92, no. 3, pp. 280–284, 2017.

[7] M. Nizami, A. Haque, P. Nguyen, and M. Hossain, "On the application of home energy management systems for power grid support," *Energy*, vol. 188, p. 116104, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544219317992

[8] C.-J. Yang and R. B. Jackson, "Opportunities and barriers to pumped-hydro energy storage in the united states," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 1, pp. 839–844, 2011.

[9] M. Amin, "Smart grid," *Public Utilities Fortnightly*, 2015.

[10] M. D. de Souza Dutra, M. F. Anjos, and S. L. Digabel, "A general framework for customized transition to smart homes," *Energy*, vol. 189, p. 116138, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036054421931833X

[11] Y. Yu, W. Luan *et al.*, "Smart grid and its implementations," *Proceedings of the CSEE*, vol. 29, no. 34, pp. 1–8, 2009.

[12] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE transactions on Industrial informatics*, vol. 7, no. 4, pp. 529–539, 2011.

[13] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE transactions on smart grid*, vol. 1, no. 1, pp. 57–64, 2010.

[14] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security & Privacy*, vol. 7, no. 3, pp. 75–77, 2009.

[15] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 81–85, 2010.

[16] A. R. Metke and R. L. Ekl, "Security technology for smart grid networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 99–107, 2010.

[17] R. Leszczyna, "A review of standards with cybersecurity requirements for smart grid," *Computers and Security*, vol. 77, pp. 262 – 276, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404818302803

[18] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner, "A Survey of Visualization Systems for Malware Analysis," in *Eurographics Conference on Visualization (EuroVis) - STARs*, 2015. [Online]. Available: https://doi.org/10.2312/eurovisstar.20151114

[19] A. González, Á. Herrero, and E. Corchado, "Neural visualization of android malware families," in *Proceedings of the International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*, 2016, pp. 574–583. [Online]. Available: https://doi.org/10.1007/978-3-319-47364-2_56

[20] A. Paturi, M. Cherukuri, J. Donahue, and S. Mukkamala, "Mobile malware visual analytics and similarities of attack toolkits (malware gene analysis)," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 149–154. [Online]. Available: https://doi.org/10.1109/CTS.2013.6567221

[21] W. Park, K. Lee, K. Cho, and W. Ryu, "Analyzing and detecting method of android malware via disassembling and visualization," in *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, 2014, pp. 817–818. [Online]. Available: https://doi.org/10.1109/ICTC.2014.6983300

[22] V. Moonsamy, J. Rong, and S. Liu, "Mining permission patterns for contrasting clean and malicious android applications," *Future Generation Computer Systems*, vol. 36, pp. 122 – 132, 2014. [Online]. Available: https://doi.org/10.1016/j.future.2013.09.014

[23] O. Somarriba, U. Zurutuza, R. Uribeetxeberria, L. Delosieres, and S. Nadjm-Tehrani, "Detection and visualization of android malware behavior," *Journal of Electrical and Computer Engineering*, vol. 2016, 2016. [Online]. Available: http://dx.doi.org/10.1155/2016/8034967

[24] Á. Herrero, U. Zurutuza, and E. Corchado, "A neural-visualization IDS for honeynet data," *Int. J. Neural Syst.*, vol. 22, no. 2, 2012. [Online]. Available: https://doi.org/10.1142/S0129065712500050

[25] Basque honeypot network. (2010 (accessed May 10, 2010)) Euskalert. [Online]. Available: http://www.euskalert.net

[26] E. Corchado and Á. Herrero, "Neural visualization of network traffic data for intrusion detection," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2042–2056, 2011. [Online]. Available: https://doi.org/10.1016/j.asoc.2010.07.002

[27] M. I. o. T. Lincoln Laboratory, *2000 DARPA Intrusion Detection Scenario Specific Datasets*, 2019 (accessed December 3, 2019). [Online]. Available: https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets

[28] Á. Herrero, M. Navarro, E. Corchado, and V. Julián, "RT-MOVICAB-IDS: addressing real-time intrusion detection," *Future Generation Comp. Syst.*, vol. 29, no. 1, pp. 250–261, 2013. [Online]. Available: https://doi.org/10.1016/j.future.2010.12.017

[29] R. Sánchez, Á. Herrero, and E. Corchado, "Visualization and clustering for SNMP intrusion detection," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 505–532, 2013. [Online]. Available: https://doi.org/10.1080/01969722.2013.803903

[30] R. Sánchez, Á. Herrero, and E. Corchado, "Clustering extension of MOVICAB-IDS to identify SNMP community searches," *Logic Journal of the IGPL*, vol. 23, no. 1, pp. 121–140, 2015. [Online]. Available: https://doi.org/10.1093/jigpal/jzu035

[31] R. Sánchez, Á. Herrero, and E. Corchado, "Clustering extension of MOVICAB-IDS to distinguish intrusions in flow-based data," *Logic Journal of the IGPL*, vol. 25, no. 1, pp. 83–102, 2017. [Online]. Available: https://doi.org/10.1093/jigpal/jzw047

[32] A. Sperotto, R. Sadre, F. E. van Vliet, and A. Pras, "A labeled data set for flow-based intrusion detection," in *IP Operations and Management, 9th IEEE International Workshop, IPOM 2009, Venice, Italy, October 29-30, 2009. Proceedings*, 2009, pp. 39–50. [Online]. Available: https://doi.org/10.1007/978-3-642-04968-2_4

[33] R. Vega Vega, H. Quintián, J. L. Calvo-Rolle, Á. Herrero, and E. Corchado, "Gaining deep knowledge of android malware families through dimensionality reduction techniques," *Logic Journal of the IGPL, Ïn press"*, 2019. [Online]. Available: http://dx.doi.org/10.1093/jigpal/jzy030

[34] R. V. Vega, H. Quintián, C. Cambra, N. Basurto, Á. Herrero, and J. L. Calvo-Rolle, "Delving into android malware families with a novel neural projection method," *Complexity*, vol. 2019, pp. 6 101 697:1–6 101 697:10, 2019. [Online]. Available: https://doi.org/10.1155/2019/6101697

[35] E. Jove, J. L. Casteleiro-Roca, H. Quintián, J. A. M. Pérez, and J. L. Calvo-Rolle, "A fault detection system based on unsupervised techniques for industrial control loops," *Expert Systems*, vol. 36, no. 4, 2019. [Online]. Available: https://doi.org/10.1111/exsy.12395

[36] E. Jove, J. L. Casteleiro-Roca, H. Quintián, J. A. M. Pérez, and J. L. Calvo-Rolle, "A new approach for system malfunctioning over an industrial system control loop based on unsupervised techniques," in *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18 - San Sebastián, Spain, June 6-8, 2018, Proceedings*, 2018, pp. 415–425. [Online]. Available: https://doi.org/10.1007/978-3-319-94120-2_40

[37] H. Quintián and E. Corchado, "Beta hebbian learning as a new method for exploratory projection pursuit," *Int. J. Neural Syst.*, vol. 27, no. 6, pp. 1–16, 2017. [Online]. Available: https://doi.org/10.1142/S0129065717500241

[38] A. Berro, S. Larabi Marie-Sainte, and A. Ruiz-Gazen, "Genetic algorithms and particle swarm optimization for exploratory projection pursuit," *Ann. Math. Artif. Intell.*, vol. 60, pp. 153–178, 10 2010.

# CAPÍTULO 7

# Conclusiones

Las principales conclusiones generales que se pueden extraer de este trabajo de investigación se relacionan en los siguientes puntos:

- Se consigue una metodología basada en métodos comunes, en la que se efectúa el análisis de una estructura interna de un conjunto de datos con diferentes ataques.
- La metodología propuesta se implementa además con métodos avanzados e incipientes con el mismo objetivo, reafirmando así la validez.
- Se implementa además una modificación de la propuesta para detectar ciberataques sobre conjuntos de datos, tanto que poseen amenazas, como otros que carecen de ellas.

Como conclusiones específicas del primer caso de estudio presentado, de acuerdo a los resultados mostrados, se puede afirmar que las técnicas de reducción de dimensionalidad son una propuesta interesante para analizar de forma visual la estructura de un conjunto de datos de alta dimensionalidad en términos generales. De forma más específica tras estudiar las familias de *malware* de Android, se puede afirmar también que este tipo de técnicas permite además obtener un conocimiento profundo sobre la naturaleza de los ataques. Gracias a las proyecciones obtenidas se identifican similitudes y diferencias entres las familias contempladas en el estudio. Como conclusión final de este primer trabajo, se puede afirmar que la identificación y caracterización del *malware* de Android es todavía un desafío que requiere de grandes esfuerzos en los próximos años.

En el segundo caso de estudio se han aplicado novedosas técnicas de aprendizaje automático a los datos de *malware* del primer trabajo presentado con el mismo objetivo. Se concluye tras el trabajo que, BHL ha superado las técnicas de proyección neuronal usadas originalmente tras ser aplicadas a los mismos datos al revelar claramente la

estructura del conjunto de datos Malgenome. Además, las características identificadas como las más importantes por los métodos basados en EPP también son resaltadas por las técnicas basadas en *Decision Trees* (DT) como relevantes para diferenciar mejor entre las familias de *malware*.

En el tercer y último trabajo presentado, la claridad obtenida por las proyecciones BHL facilita la distinción entre el tráfico normal y los patrones de tráfico anómalos, lo que hace más fácil la detección temprana de ataques. Además, BHL puede identificar fácilmente en base al puerto en el que se producen los ataques, el tipo de ataque del que se trata. Remarcar, que cuando este tipo de ataque se combina con el ataque MIB, BHL no solo puede distinguir entre ellos, sino que también entre los dos tipos de ataque MIB y los puertos escaneados. Esto convierte a BHL en una poderosa herramienta para los protocolos de administración de redes. Los resultados del experimento realizado han demostrado que el rendimiento de BHL es superior al de las técnicas utilizadas en investigaciones anteriores, demostrando proyecciones fácilmente identificables, donde los ataques se distinguen claramente sobre el comportamiento normal de la red, incluso cuando se producen diferentes tipos de ataques al mismo tiempo.

# CAPÍTULO 8

# Trabajos futuros

Son múltiples las posibilidades de estudio que se pueden abordar a partir de los logros alcanzados en el presente trabajo de investigación. Han surgido, además, gran cantidad de ideas a partir de las experiencias, consecuencia de aplicación de los diferentes métodos contemplados en la propuesta.

De forma más concreta, como continuidad de esta tesis doctoral, se relacionan a continuación algunas de las diversas líneas a abordar en el futuro:

- Experimentar alternativas o nuevos métodos, no contemplados, de clasificación y detección de variables significativas, que contribuyan a conseguir mejores resultados que los del presente trabajo. Incluso con los métodos empleados, se contempla la posibilidad de conseguir mejoras incorporando nuevas variables de las que se pueda disponer de los diferentes casos de estudio.
- Comparar cómo influye en el método descrito la utilización de diferentes algoritmos de agrupamiento como fase posterior a la reducción dimensional (K-means, DBSCAN, EM-GMM, AHC, etc.).
- Descubrir y desarrollar nuevos métodos que contemplen otro tipo de obtención de reducción dimensional, con el fin de complementar y hacer la propuesta más precisa y fiable.
- Teniendo en cuenta que los sistemas reales son cambiantes en el tiempo, se plantea establecer una metodología para la creación de modelos de detección de amenazas *Ad-Hoc*, que sirvan de base para obtener una mejor identificación y categorización.
- Por la misma razón que el punto anterior, se plantea estudiar la implementación de alguna metodología adaptativa, que permita mejorar la identificación y categorización de amenazas en el tiempo.

- Estudiar la posibilidad de creación de una base de conocimiento para dar apoyo a la nueva casuística. La base se iría formando con los modelos generados para cada situación concreta y que pueda tener cierta similitud en casos semejantes.

# Justificantes de los artículos

Se incluyen a continuación las portadas de los artículos presentados en los capítulos 4, 5 y 6.

# Gaining deep knowledge of Android malware families through dimensionality reduction techniques

RAFAEL VEGA VEGA[*], HÉCTOR QUINTIÁN[**],
AND JOSÉ LUÍS CALVO-ROLLE[†], *Department of Industrial Engineering, University of A Coruña, Spain Avda. 19 de febrero S/N 15.405, Ferrol - Coruña, Spain.*

ÁLVARO HERRERO[††], *Department of Civil Engineering, University of Burgos, Spain Avenida de Cantabria s/n, 09006 Burgos, Spain.*

EMILIO CORCHADO[§], *Departamento de Informática y Automática, Universidad de Salamanca Plaza de la Merced, s/n, 37008 Salamanca, Spain.*

## Abstract

This research proposes the analysis and subsequent characterisation of Android malware families by means of low dimensional visualisations using dimensional reduction techniques. The well-known Malgenome data set, coming from the Android Malware Genome Project, has been thoroughly analysed through the following six dimensionality reduction techniques: Principal Component Analysis, Maximum Likelihood Hebbian Learning, Cooperative Maximum Likelihood Hebbian Learning, Curvilinear Component Analysis, Isomap and Self Organizing Map. Results obtained enable a clear visual analysis of the structure of this high-dimensionality data set, letting us gain deep knowledge about the nature of such Android malware families. Interesting conclusions are obtained from the real-life data set under analysis.

*Keywords*: Android malware, malware families, dimensionality reduction, artificial neural networks.

## 1 Introduction

Since the first smartphones came into the market in the late 1990s, sales on that sector have increased constantly to the present day. Among all the available operating systems, Google's Android has been, and increasingly is, the most popular mobile platform [1]. The number of Android units sold in Q1 2017 worldwide raised to 379.98 million of 432.79 million units, that is a share of 87.79%. It is not only the number of devices but also the number of apps—those available at Google Play (Android's official store) constantly increase, up to more than 3.4 million are available nowadays [2]. With regard to the security issue, the number of malicious Android apps has greatly risen in the past four years;

[*]E-mail: ravega@udc.es
[**]E-mail: hector.quintian@udc.es
[†]E-mail: jlcalvo@udc.es
[††]E-mail: ahcosio@ubu.es
[§]E-mail: escorchado@usal.es

Vol. 27, No. 2, © The Author(s) 2018. Published by Oxford University Press. 
Advance Access published 11 September 2018 doi:10.1093/jigpal/jzy030

Hindawi
Complexity
Volume 2019, Article ID 6101697, 10 pages
https://doi.org/10.1155/2019/6101697

*Research Article*

# Delving into Android Malware Families with a Novel Neural Projection Method

**Rafael Vega Vega**,[1] **Héctor Quintián**,[1] **Carlos Cambra**,[2] **Nuño Basurto**,[2] **Álvaro Herrero**,[2] **and José Luis Calvo-Rolle**[1,3]

[1]*University of A Coruña, Departamento de Ingeniería Industrial, Avda. 19 de febrero s/n, 15495, Ferrol, A Coruña, Spain*
[2]*Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Civil, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006, Burgos, Spain*
[3]*Research Institute of Applied Sciences in Cybersecurity (RIASC), Spain*

Correspondence should be addressed to Rafael Vega Vega; rafael.alejandro.vega.vega@udc.es

Received 5 December 2018; Accepted 23 January 2019; Published 2 June 2019

Guest Editor: Alicja Krzemień



Present research proposes the application of unsupervised and supervised machine-learning techniques to characterize Android malware families. More precisely, a novel unsupervised neural-projection method for dimensionality-reduction, namely, Beta Hebbian Learning (BHL), is applied to visually analyze such malware. Additionally, well-known supervised Decision Trees (DTs) are also applied for the first time in order to improve characterization of such families and compare the original features that are identified as the most important ones. The proposed techniques are validated when facing real-life Android malware data by means of the well-known and publicly available Malgenome dataset. Obtained results support the proposed approach, confirming the validity of BHL and DTs to gain deep knowledge on Android malware.

## 1. Introduction and Previous Work

Undoubtedly, smartphones are one of the emerging technologies that have revolutionized the use of computing systems. From the very beginning (late 1990s), more and more smartphones are sold every year and it is expected that the number of smartphone users passes the 2.7 billion mark by 2019 [1]. Although there is a variety of operating systems for such devices, Google's Android is the most widely used one [1] and, consequently, the number of Android users has permanently increased. Concurrently, the number of Android apps strongly increased in the last years but it started to decline from 3.6 million in March, 2017 (highest value), to 2.6 million in September, 2018 [2].

From the security standpoint, one of the main problems of smartphone apps is malware that is included in software in general and in these apps in particular. Furthermore, "users of mobile devices are increasingly subject to malicious activity pushing malware apps" [3]. It is true that some effort has been devoted by Google to remove and prevent malicious apps from Google Play Market, but malware is still there [3]. Moreover, malware Android apps are increasing; in the third trimester of 2018 there has been an increase of 1.7 million detections [4].

As it can be seen, privacy and security of smartphones still are open challenges [5] and many researchers are working on this topic. To better fight against malware and be able to develop an effective solution, understanding it and its nature is required [6]. In keeping with this idea, present paper proposes getting deeper knowledge about Android malware for its better detection. More precisely, both supervised (Decision Trees) and unsupervised (Neural Projection Method) machine-learning techniques are applied to increase our knowledge about the main families of Android malware. In order to validate the proposed techniques, they are applied to the well-known Malgenome dataset [7] that is open and real-life.

This pioneering work on collecting Android malware found some interesting statistics [6] motivating further analysis of malware: 36.7% of the collected apps leverage root-level

applied sciences

MDPI

Article

# Intrusion Detection with Unsupervised Techniques for Network Management Protocols over Smart Grids

**Rafael Alejandro Vega Vega [1,†], Pablo Chamoso-Santos [2,3,†], Alfonso González Briones [2,3,4,†], José-Luis Casteleiro-Roca [1,†], Esteban Jove [1,†], María del Carmen Meizoso-López [1,†], Benigno Antonio Rodríguez-Gómez [1,†], Héctor Quintián [1,†,*], Álvaro Herrero [5,†], Kenji Matsui [6,†] and Emilio Corchado [2,†] and José Luis Calvo-Rolle [1,†]**

1 Department of Industrial Engineering, University of A Coruña, 15403 Ferrol, Spain; rafael.alejandro.vega.vega@udc.es (R.A.V.V.); jose.luis.casteleiro@udc.es (J.-L.C.-R.); esteban.jove@udc.es (E.J.); carmen.meizoso@udc.es (M.d.C.M.-L.); benigno.rodriguez@udc.es (B.A.R.-G.); jlcalvo@udc.es (J.L.C.-R.)
2 BISITE Research Group, University of Salamanca, Edificio I+D+i, Calle Espejo 2, 37007 Salamanca, Spain; chamoso@usal.es (P.C.-S.); alfonsogb@usal.es (A.G.B.); escorchado@usal.es (E.C.)
3 Air Institute, IoT Digital Innovation Hub (Spain), Calle Segunda 4, 37188 Salamanca, Spain
4 Research Group on Agent-Based, Social and Interdisciplinary Applications (GRASIA), Complutense University of Madrid, 28040 Madrid, Spain
5 Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain; ahcosio@ubu.es
6 Faculty of Robotics & Design, Osaka Institute of Technology, Osaka 535-8585, Japan; kenji.matsui@oit.ac.jp
* Correspondence: hector.quintian@udc.es; Tel.: +34-881013117
† These authors contributed equally to this work.

Received: 14 January 2020; Accepted: 22 March 2020; Published: 27 March 2020

**Abstract:** The present research work focuses on overcoming cybersecurity problems in the Smart Grid. Smart Grids must have feasible data capture and communications infrastructure to be able to manage the huge amounts of data coming from sensors. To ensure the proper operation of next-generation electricity grids, the captured data must be reliable and protected against vulnerabilities and possible attacks. The contribution of this paper to the state of the art lies in the identification of cyberattacks that produce anomalous behaviour in network management protocols. A novel neural projectionist technique (Beta Hebbian Learning, BHL) has been employed to get a general visual representation of the traffic of a network, making it possible to identify any abnormal behaviours and patterns, indicative of a cyberattack. This novel approach has been validated on 3 different datasets, demonstrating the ability of BHL to detect different types of attacks, more effectively than other state-of-the-art methods.

**Keywords:** smart grid; computational intelligence; automatic response; exploratory projection pursuit; neural networks

## 1. Introduction

Care for the environment is not a simple trend. It is a very important matter from a legal point of view. Governments have already implemented regulations, making it compulsory to take action against environmental degradation, and there will certainly be more regulations in the future. It is necessary to remark that zero impact is impossible from a practical point of view. Nevertheless, it is necessary to pursue sustainability and to minimize impact [1]. Renewable energy systems play a very

# Publicaciones del doctorando

En este capítulo se recogen las publicaciones del doctorando. Se presentan por orden cronológico las publicaciones científicas indexadas en el JCR

1. Gaining deep knowledge of Android malware families through dimensionality reduction techniques [1].
2. Delving into Android malware families with a novel neural projection method [2].
3. A hybrid intelligent system to forecast solar energy production [3].
4. Intrusion detection with unsupervised techniques for network management protocols over Smart Grids [4].
5. Solar thermal collector output temperature prediction by hybrid intelligent model for Smartgrid and Smartbuildings applications and optimization [5].
6. Hybrid intelligent model to predict the Remifentanil infusion rate in patients under general anesthesia [6].

## Bibliografía


[1] R. Vega Vega, H. Quintián, J. L. Calvo-Rolle, Á. Herrero, and E. Corchado, "Gaining deep knowledge of android malware families through dimensionality reduction techniques," *Logic Journal of the IGPL*, vol. 27, no. 2, pp. 160–176, 2019.

[2] R. Vega Vega, H. Quintián, C. Cambra, N. Basurto, Á. Herrero, and J. L. Calvo-Rolle, "Delving into android malware families with a novel neural projection method," *Complexity*, vol. 2019, 2019.

[3] N. Basurto, Á. Arroyo, R. Vega, H. Quintián, J. L. Calvo-Rolle, and Á. Herrero, "A hybrid intelligent system to forecast solar energy production," *Computers & Electrical Engineering*, vol. 78, pp. 373–387, 2019.

[4] R. A. Vega Vega, P. Chamoso-Santos, A. G. Briones, J.-L. Casteleiro-Roca, E. Jove, M. del Carmen Meizoso-López, B. A. Rodríguez-Gómez, H. Quintián, Á. Herrero, K. Matsui *et al.*, "Intrusion detection with unsupervised techniques for network management protocols over smart grids," *Applied Sciences*, vol. 10, no. 7, p. 2276, 2020.

[5] J.-L. Casteleiro-Roca, P. Chamoso, E. Jove, A. González-Briones, H. Quintián, M.-I. Fernández-Ibáñez, R. A. Vega Vega, A.-J. Piñón Pazos, J. A. López Vázquez, S. Torres-Álvarez *et al.*, "Solar thermal collector output temperature prediction by hybrid intelligent model for smartgrid and smartbuildings applications and optimization," *Applied Sciences*, vol. 10, no. 13, p. 4644, 2020.

[6] E. Jove, J. M. Gonzalez-Cava, J.-L. Casteleiro-Roca, H. Quintián, J. A. Méndez Pérez, R. Vega Vega, F. Zayas-Gato, F. J. de Cos Juez, A. León, M. MartÍn *et al.*, "Hybrid intelligent model to predict the remifentanil infusion rate in patients under general anesthesia," *Logic Journal of the IGPL*, 2020.

# Referencias


[1] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014.

[2] D. Craigen, N. Diakun-Thibault, and R. Purse, "Defining cybersecurity," *Technology Innovation Management Review*, vol. 4, no. 10, 2014.

[3] P. W. Singer and A. Friedman, *Cybersecurity: What everyone needs to know*. oup usa, 2014.

[4] M. Uma and G. Padmavathi, "A survey on various cyber attacks and their classification." *Int. J. Netw. Secur.*, vol. 15, no. 5, pp. 390–396, 2013.

[5] A. M. Shabut, K. T. Lwin, and M. A. Hossain, "Cyber attacks, countermeasures, and protection schemes—a state of the art survey," in *2016 10th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. IEEE, 2016, pp. 37–44.

[6] C.-T. Lin, S.-L. Wu, and M.-L. Lee, "Cyber attack and defense on industry control systems," in *2017 IEEE Conference on Dependable and Secure Computing*. IEEE, 2017, pp. 524–526.

[7] C. J. Bartodziej, "The concept industry 4.0," in *The concept industry 4.0*. Springer, 2017, pp. 27–50.

[8] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE industrial electronics magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[9] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business and information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[10] L. Thames and D. Schaefer, "Software-defined cloud manufacturing for industry 4.0," *Procedia CIRP*, vol. 52, pp. 12–17, 2016, the Sixth International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2016).

[11] J. C. Bendul and H. Blunck, "The design space of production planning and control for industry 4.0," *Computers in Industry*, vol. 105, pp. 260–272, 2019.

[12] A. Rojko, "Industry 4.0 concept: Background and overview." *International Journal of Interactive Mobile Technologies*, vol. 11, no. 5, 2017.

[13] J. Tupa, J. Simota, and F. Steiner, "Aspects of risk management implementation for industry 4.0," *Procedia manufacturing*, vol. 11, pp. 1223–1230, 2017.

[14] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *Ieee Access*, vol. 6, pp. 3585–3593, 2018.

[15] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, "Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Annals of operations research*, pp. 1–19, 2020.

[16] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *Journal of Big data*, vol. 7, no. 1, pp. 1–29, 2020.

[17] M. Lezzi, M. Lazoi, and A. Corallo, "Cybersecurity for industry 4.0 in the current literature: A reference framework," *Computers in Industry*, vol. 103, pp. 97–110, 2018.

[18] T. Nguyen, R. G. Gosine, and P. Warrian, "A systematic review of big data analytics for oil and gas industry 4.0," *IEEE Access*, vol. 8, pp. 61 183–61 201, 2020.

[19] R. Gupta, S. Tanwar, N. Kumar, and S. Tyagi, "Blockchain-based security attack resilience schemes for autonomous vehicles in industry 4.0: A systematic review," *Computers & Electrical Engineering*, vol. 86, p. 106717, 2020.

[20] N. S. Arunraj, R. Hable, M. Fernandes, K. Leidl, and M. Heigl, "Comparison of supervised, semi-supervised and unsupervised learning methods in network intrusion detection system (nids) application," *Anwendungen und Konzepte der Wirtschaftsinformatik*, vol. 6, 2017.

[21] C. Alcaraz, "Secure interconnection of it-ot networks in industry 4.0," in *Critical Infrastructure Security and Resilience*. Springer, 2019, pp. 201–217.

[22] G. Culot, F. Fattori, M. Podrecca, and M. Sartor, "Addressing industry 4.0 cybersecurity challenges," *IEEE Engineering Management Review*, vol. 47, no. 3, pp. 79–86, 2019.

[23] M. N. Al-Mhiqani, R. Ahmad, W. Yassin, A. Hassan, Z. Z. Abidin, N. S. Ali, and K. H. Abdulkareem, "Cyber-security incidents: a review cases in cyber-physical systems," *Int. J. Adv. Comput. Sci. Appl*, no. 1, pp. 499–508, 2018.

[24] M. Chiarvesio and R. Romanello, "Industry 4.0 technologies and internationalization: Insights from italian companies," in *International Business in the Information and Digital Age*. Emerald Publishing Limited, 2018.

[25] J. P. Vilko and J. M. Hallikas, "Risk assessment in multimodal supply chains," *International Journal of Production Economics*, vol. 140, no. 2, pp. 586–595, 2012.

[26] Z. Rajnai and I. Kocsis, "Assessing industry 4.0 readiness of enterprises," in *2018 IEEE 16th world symposium on applied machine intelligence and informatics (SAMI)*. IEEE, 2018, pp. 000 225–000 230.

[27] M. D. Cavelty and F. J. Egloff, "The politics of cybersecurity: Balancing different roles of the state," *St Antony's International Review*, vol. 15, no. 1, pp. 37–57, 2019.

[28] B. Farrand and H. Carrapico, "Blurring public and private: cybersecurity in the age of regulatory capitalism," in *Security Privatization*. Springer, 2018, pp. 197–217.

[29] H. N. Alshabib and J. T. Martins, "Cybersecurity: Perceived threats and policy responses in the gulf cooperation council," *IEEE Transactions on Engineering Management*, 2021.

[30] S. Petrenko, K. Makoveichuk, and A. Olifirov, "New methods of the cybersecurity knowledge management analytics," in *International Conference on Convergent Cognitive Information Technologies*. Springer, 2018, pp. 296–310.

[31] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain for 5g-enabled iot for industrial automation: A systematic review, solutions, and challenges," *Mechanical Systems and Signal Processing*, vol. 135, p. 106382, 2020.

[32] A. Ahmadi, M. Moradi, C. Cherifi, V. Cheutet, and Y. Ouzrout, "Wireless connectivity of cps for smart manufacturing: A survey," in *2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*. IEEE, 2018, pp. 1–8.

[33] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[34] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial iot: Blockchain system with credit-based consensus mechanism," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3680–3689, 2019.

[35] Statista. Smartphone sales by os worldwide 2009-2017. [Online]. Available: https://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operatingsystem/

[36] AppBrain. Android operating system statistics. [Online]. Available: https://www.appbrain.com/stats/stats-index

[37] SOPHOSLABS, "Sophoslabs 2018 malware forecast," Tech. Rep., 2017.

[38] Malwarebytes, "Malwarebytes labs report: Cybercrime tactics and techniques q3 2017," Tech. Rep., 2017.

[39] M. Proyect. Android malware genome project. [Online]. Available: http://www.malgenomeproject.org

[40] R. F. Garcia, J. L. C. Rolle, M. R. Gomez, and A. D. Catoira, "Expert condition monitoring on hydrostatic self-levitating bearings," *Expert Systems with Applications*, vol. 40, no. 8, pp. 2975–2984, 2013.

[41] L. Sáiz, A. Pérez, Á. Herrero, and E. Corchado, "Analyzing key factors of human resources management," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2011, pp. 463–473.

[42] C. C. Turrado, M. d. C. M. López, F. S. Lasheras, B. A. R. Gómez, J. L. C. Rollé, and F. J. d. C. Juez, "Missing data imputation of solar radiation data under different atmospheric conditions," *Sensors*, vol. 14, no. 11, pp. 20 382–20 399, 2014.

[43] I. J. Machón González, H. López García, and J. L. Calvo Rolle, "Neuro-robust controller for non-linear systems (controlador neurorobusto para sistemas no lineales)," *Dyna*, 2011.

[44] E. Corchado, Á. Herrero, and J. M. Sáiz, "Testing cab-ids through mutations: on the identification of network scans," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2006, pp. 433–441.

[45] C. I. Pinzon, J. F. De Paz, A. Herrero, E. Corchado, J. Bajo, and J. M. Corchado, "idmas-sql: intrusion detection based on mas to detect and block sql injection through data mining," *Information Sciences*, vol. 231, pp. 15–31, 2013.

[46] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, J. Nieves, P. G. Bringas, and G. Álvarez Marañón, "Mama: manifest analysis for malware detection in android," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 469–488, 2013.

[47] E. Corchado, D. MacDonald, and C. Fyfe, "Maximum and minimum likelihood hebbian learning for exploratory projection pursuit," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 203–225, 2004.

[48] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[49] E. Oja, "Principal components, minor components, and linear neural networks," *Neural networks*, vol. 5, no. 6, pp. 927–935, 1992.

[50] C. Fyfe, "A neural network for pca and beyond," *Neural Processing Letters*, vol. 6, no. 1-2, pp. 33–41, 1997.

[51] C. Fyfe, D. R. McGregor, and R. Baddeley, *Exploratory projection pursuit: an artificial neural network approach*. Department of Computer Science, University of Strathclyde, 1994.

[52] E. Corchado and C. Fyfe, "Connectionist techniques for the identification and suppression of interfering underlying factors," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 08, pp. 1447–1466, 2003.

[53] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. I–I.

[54] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on neural networks*, vol. 8, no. 1, pp. 148–154, 1997.

[55] G. Cirrincione, J. Hérault, and V. Randazzo, "The on-line curvilinear component analysis (oncca) for real-time data reduction," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[56] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

[57] N. Chen, B. Ribeiro, A. Vieira, and A. Chen, "Clustering and visualization of bankruptcy trajectory using self-organizing map," *Expert Systems with Applications*, vol. 40, no. 1, pp. 385–393, 2013.

[58] J. J. Fuertes, M. Domínguez, P. Reguera, M. A. Prada, I. Díaz, and A. A. Cuadrado, "Visual dynamic model based on self-organizing maps for supervision and fault detection in industrial processes," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 1, pp. 8–17, 2010.

[59] T. Kohonen, "The self-organizing map. neurocomputing 21, 1e6," 1998.

[60] E. Mohebi and A. Bagirov, "Constrained self organizing maps for data clusters visualization," *Neural Processing Letters*, vol. 43, no. 3, pp. 849–869, 2016.

[61] Y. Wu, T. K. Doyle, and C. Fyfe, "Multi-layer topology preserving mapping for k-means clustering," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2011, pp. 84–91.

[62] H. Quintián and E. Corchado, "Beta scale invariant map," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 218–235, 2017.

[63] H. FAKHOURI, L. CHERRAT, and M. Ezziyyani, "Towards a new approach to improve the classification accuracy of the kohonen's self-organizing map during learning process."

[64] T. Kohonen, "Essentials of the self-organizing map," *Neural networks*, vol. 37, pp. 52–65, 2013.

[65] Gartner. (2018) Global smartphone sales to end users from 1st quarter 2009. [Online]. Available: https://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/

[66] AppBrain. (2018) Android and google play statistics. [Online]. Available: https://www.appbrain.com/stats/stats-index

[67] SOPHOSLABS, "Ltd., s., sophoslabs 2019 threat report," Tech. Rep., 2019.

[68] M. LABS, "Labs, m., cybercrime tactics and techniques: Q3 2018," Tech. Rep., 2018.

[69] S. Arshad, M. A. Shah, A. Khan, and M. Ahmed, "Android malware detection & protection: A survey," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, 2016.

[70] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2012, pp. 95–109.

[71] Y. Zhou. (2010) Malgenome project. [Online]. Available: http://www.malgenomeproject.org

[72] J. Sedano, S. González, C. Chira, Á. Herrero, E. Corchado, and J. R. Villar, "Key features for the characterization of android malware families," *Logic Journal of the IGPL*, vol. 25, no. 1, pp. 54–66, 2017.

[73] S. L. Marie-Sainte, "Detection and visualization of non-linear structures in large datasets using exploratory projection pursuit laboratory (epp-lab) software," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 1, pp. 2 – 18, 2017.

[74] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[75] P. G. Nieto, J. M. Torres, F. de Cos Juez, and F. S. Lasheras, "Using multivariate adaptive regression splines and multilayer perceptron networks to evaluate paper manufactured using eucalyptus globulus," *Applied Mathematics and Computation*, vol. 219, no. 2, pp. 755 – 763, 2012.

[76] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," *Expert Systems with Applications*, vol. 36, no. 1, pp. 2 – 17, 2009.

[77] R. Ferreiro García, J. L. Calvo Rolle, M. Romero Gómez, and A. De Miguel Catoira, "Expert condition monitoring on hydrostatic self-levitating bearings," *Expert Systems with Applications*, vol. 40, no. 8, pp. 2975 – 2984, 2013.

[78] C. Crespo Turrado, M. d. C. Meizoso López, F. Sánchez Lasheras, B. A. Rodríguez Gómez, J. L. Calvo Rolle, and F. J. De Cos Juez, "Missing data imputation of solar radiation data under different atmospheric conditions," *Sensors*, vol. 14, no. 11, pp. 20 382–20 399, 2014.

[79] J. L. Calvo Rolle, I. Machón González, and H. López García, "Neuro-robust controller for non-linear systems," *Dyna*, vol. 86, no. 3, pp. 308–317, 2011.

[80] Á. Herrero, E. Corchado, M. A. Pellicer, and A. Abraham, "Hybrid multi agent-neural network intrusion detection with mobile visualization," in *Innovations in Hybrid Intelligent Systems*, 2008, pp. 320–328.

[81] R. Sánchez, Á. Herrero, and E. Corchado, "Visualization and clustering for SNMP intrusion detection," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 505–532, 2013.

[82] C. Pinzón, Á. Herrero, J. F. de Paz, E. Corchado, and J. Bajo, "Cbrid4sql: A CBR intrusion detector for SQL injection attacks," in *Proceedings of the 5th International Conference on Hybrid Artificial Intelligence Systems HAIS 2010 - Part II*, 2010, pp. 510–519.

[83] C. Pinzón, J. F. de Paz, J. Bajo, Á. Herrero, and E. Corchado, "Aiida-sql: An adaptive intelligent intrusion detector agent for detecting sql injection attacks," in *Proceedings of the 10th International Conference on Hybrid Intelligent Systems HIS 2010*, 2010, pp. 73–78.

[84] Á. Herrero, U. Zurutuza, and E. Corchado, "A neural-visualization IDS for honeynet data," *Int. J. Neural Syst.*, vol. 22, no. 2, 2012.

[85] D. Atienza, Á. Herrero, and E. Corchado, "Neural analysis of HTTP traffic for web attack detection," in *Proceedings of the 8th International Conference on Computational Intelligence in Security for Information Systems CISIS 2015*, 2015, pp. 201–212.

[86] L. Cen, C. S. Gates, L. Si, and N. Li, "A probabilistic discriminative model for android malware detection with decompiled source code," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 400–412, 2015.

[87] H.-J. Zhu, Z.-H. You, Z.-X. Zhu, W.-L. Shi, X. Chen, and L. Cheng, "Droiddet: Effective and robust detection of android malware using static analysis along with rotation forest model," *Neurocomputing*, vol. 272, pp. 638 – 646, 2018.

[88] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," *Soft Computing*, vol. 20, no. 1, pp. 343–357, 2016.

[89] A. Paturi, M. Cherukuri, J. Donahue, and S. Mukkamala, "Mobile malware visual analytics and similarities of attack toolkits (malware gene analysis)," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 149–154.

[90] W. Park, K. Lee, K. Cho, and W. Ryu, "Analyzing and detecting method of android malware via disassembling and visualization," in *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, 2014, pp. 817–818.

[91] V. Moonsamy, J. Rong, and S. Liu, "Mining permission patterns for contrasting clean and malicious android applications," *Future Generation Computer Systems*, vol. 36, pp. 122 – 132, 2014.

[92] S. Singh and P. Gupta, "Comparative study id3, cart and c4. 5 decision tree algorithm: a survey," *International Journal of Advanced Information Science and Technology*, vol. 27, no. 7, pp. 97–103, 2014.

[93] W.-Y. Loh and Y.-S. Shih, "Split selection methods for classification trees," *Statistica sinica*, vol. 7, no. 4, pp. 815–840, 1997.

[94] W.-Y. Loh, "Regression trees with unbiased variable selection and interaction detection," *Statistica Sinica*, vol. 12, no. 12, pp. 361–386, 2002.

[95] P. Teufl, M. Ferk, A. Fitzek, D. Hein, S. Kraxberger, and C. Orthacker, "Malware detection by applying knowledge discovery processes to application metadata on the android market (google play)," *Security and Communication Networks*, vol. 9, no. 5, pp. 389–419, 2016.

[96] J.-w. Jang, J. Yun, A. Mohaisen, J. Woo, and H. K. Kim, "Detecting and classifying method based on similarity matching of android malware behavior with profile," *SpringerPlus*, vol. 5, no. 1, p. 273, Mar 2016.

[97] A. Altaher, "An improved android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (ehnfc) and permission-based features," *Neural Computing and Applications*, vol. 28, no. 12, pp. 4147–4157, Dec 2017.

[98] L. Li, A. Bartel, T. F. BissyandÃ©, J. Klein, Y. L. Traon, S. Arzt, S. Rasthofer, E. Bodden, D. Octeau, and P. McDaniel, "Iccta: Detecting inter-component privacy leaks in android apps," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 1, 2015, pp. 280–291.

[99] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket," in *2014 Network and Distributed System Security (NDSS) Symposium*, vol. 14, 2014, pp. 23–26.

[100] G. Suarez-Tangil, S. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "Droidsieve: Fast and accurate classification of obfuscated android malware," in *Seventh ACM on Conference on Data and Application Security and Privacy (CODASPY)*, 2017.

[101] J. Sedano, C. Chira, S. González, Á. Herrero, E. Corchado, and J. R. Villar, "On the selection of key features for android malware characterization," in *International Joint Conference - CISIS'15 and ICEUTE'15, 8th International Conference on Computational Intelligence in Security for Information Systems / 6th International Conference on EUropean Transnational Education, Burgos, Spain, 15-17 June, 2015*, 2015, pp. 167–176.

[102] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner, "A Survey of Visualization Systems for Malware Analysis," in *Eurographics Conference on Visualization (EuroVis) - STARs*, 2015.

[103] A. González, Á. Herrero, and E. Corchado, "Neural visualization of android malware families," in *Proceedings of the International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*, 2016, pp. 574–583.

[104] O. Somarriba, U. Zurutuza, R. Uribeetxeberria, L. Delosieres, and S. Nadjm-Tehrani, "Detection and visualization of android malware behavior," *Journal of Electrical and Computer Engineering*, vol. 2016, 2016.

[105] L. Sánchez-González, L. Fernández-Robles, M. C. Limas, J. Alfonso-Cendón, H. Pérez, H. Quintián, and E. Corchado, "Use of classifiers and recursive feature elimination to assess boar sperm viability," *Logic Journal of the IGPL*, vol. 26, no. 6, pp. 629–637, 2018.

[106] E. Corchado and Á. Herrero, "Neural visualization of network traffic data for intrusion detection," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2042–2056, 2011.

[107] M. I. o. T. Lincoln Laboratory, *2000 DARPA Intrusion Detection Scenario Specific Datasets*, 2019 (accessed December 3, 2019). [Online]. Available: https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets

[108] basque honeypot network, *Euskalert*, 2010 (accessed May 10, 2010). [Online]. Available: http://www.euskalert.net

[109] Á. Herrero, M. Navarro, E. Corchado, and V. Julián, "RT-MOVICAB-IDS: addressing real-time intrusion detection," *Future Generation Comp. Syst.*, vol. 29, no. 1, pp. 250–261, 2013.

[110] R. Sánchez, Á. Herrero, and E. Corchado, "Clustering extension of MOVICAB-IDS to identify SNMP community searches," *Logic Journal of the IGPL*, vol. 23, no. 1, pp. 121–140, 2015.

[111] R. Sánchez, Á. Herrero, and E. Corchado, "Clustering extension of MOVICAB-IDS to distinguish intrusions in flow-based data," *Logic Journal of the IGPL*, vol. 25, no. 1, pp. 83–102, 2017.

[112] A. Sperotto, R. Sadre, F. E. van Vliet, and A. Pras, "A labeled data set for flow-based intrusion detection," in *IP Operations and Management, 9th IEEE International Workshop, IPOM 2009, Venice, Italy, October 29-30, 2009. Proceedings*, 2009, pp. 39–50.

[113] R. Vega Vega, H. Quintián, J. L. Calvo-Rolle, Á. Herrero, and E. Corchado, "Gaining deep knowledge of android malware families through dimensionality reduction techniques," *Logic Journal of the IGPL, Ïn press"*, 2019.

[114] H. Quintián and E. Corchado, "Beta scale invariant map," *Eng. Appl. of AI*, vol. 59, pp. 218–235, 2017.

[115] E. Jove, J. L. Casteleiro-Roca, H. Quintián, J. A. M. Pérez, and J. L. Calvo-Rolle, "A fault detection system based on unsupervised techniques for industrial control loops," *Expert Systems*, vol. 36, no. 4, 2019.

[116] E. Jove, J. L. Casteleiro-Roca, H. Quintián, J. A. M. Pérez, and J. L. Calvo-Rolle, "A new approach for system malfunctioning over an industrial system control loop based on unsupervised techniques," in *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18 - San Sebastián, Spain, June 6-8, 2018, Proceedings*, 2018, pp. 415–425.

[117] H. Quintián and E. Corchado, "Beta hebbian learning as a new method for exploratory projection pursuit," *Int. J. Neural Syst.*, vol. 27, no. 6, pp. 1–16, 2017.

[118] T. Kuwae and M. Hori, "Global environmental issues," *Blue Carbon in Shallow Coastal Ecosystems: Carbon Dynamics, Policy, and Implementation*, 2019.

[119] H. Karunathilake, K. Hewage, W. Mérida, and R. Sadiq, "Renewable energy selection for net-zero energy communities: Life cycle based decision making under uncertainty," *Renewable energy*, vol. 130, pp. 558–573, 2019.

[120] R. Prakash, I. K. Bhat *et al.*, "Energy, economics and environmental impacts of renewable energy systems," *Renewable and sustainable energy reviews*, vol. 13, no. 9, pp. 2716–2721, 2009.

[121] "Energy footprint controlled by urban demands: How much does supply chain complexity contribute?" *Energy*, vol. 183, pp. 561 – 572, 2019.

[122] J.-A. Montero-Sousa, J.-L. Casteleiro-Roca, and J.-L. Calvo-Rolle, "Evolution of the electricity sector after the 2nd world war," *DYNA*, vol. 92, no. 3, pp. 280–284, 2017.

[123] C.-J. Yang and R. B. Jackson, "Opportunities and barriers to pumped-hydro energy storage in the united states," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 1, pp. 839–844, 2011.

[124] "On the application of home energy management systems for power grid support," *Energy*, vol. 188, p. 116104, 2019.

[125] "A general framework for customized transition to smart homes," *Energy*, vol. 189, p. 116138, 2019.

[126] A. Berro, S. Larabi Marie-Sainte, and A. Ruiz-Gazen, "Genetic algorithms and particle swarm optimization for exploratory projection pursuit," *Ann. Math. Artif. Intell.*, vol. 60, pp. 153–178, 10 2010.

[127] S. Hou and P. D. Wentzell, "Re-centered kurtosis as a projection pursuit index for multivariate data analysis," *Journal of Chemometrics*, vol. 28, no. 5, pp. 370–384, 2014.

[128] J. Friedman and J. Tukey, "A projection pursuit algorithm for exploratory data analysis, ieee transactions on computers, c-23, 881-889," *Computers, IEEE Transactions on*, vol. C 23, pp. 881 – 890, 10 1974.

[129] E. Oja, "Neural networks, principal components, and subspaces," *Int. J. Neural Syst.*, vol. 1, no. 1, pp. 61–68, 1989.

[130] Y. Yu, W. Luan *et al.*, "Smart grid and its implementations," *Proceedings of the CSEE*, vol. 29, no. 34, pp. 1–8, 2009.

[131] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE transactions on Industrial informatics*, vol. 7, no. 4, pp. 529–539, 2011.

[132] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE transactions on smart grid*, vol. 1, no. 1, pp. 57–64, 2010.

[133] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security & Privacy*, vol. 7, no. 3, pp. 75–77, 2009.

[134] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 81–85, 2010.

[135] A. R. Metke and R. L. Ekl, "Security technology for smart grid networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 99–107, 2010.