Beta Hebbian Learning for intrusion detection in networks with MQTT Protocols for IoT devices

ÁLVARO MICHELENA[*], *Department of Industrial Engineering, University of A Coruña, CTC, CITIC, 15403 Ferrol, A Coruña, Spain.*

MARÍA TERESA GARCÍA ORDÁS[**], *Department of Electrical and Systems Engineering, University of León, 24007 León, Spain.*

JOSÉ AVELEIRA-MATA[†], *Department of Electrical and Systems Engineering, University of León, 24007 León, Spain.*

DAVID YEREGUI MARCOS DEL BLANCO[††], *Department of Mechanic Engineering, Computer and Aerospacial Sciences, University of León, 24007 León, Spain.*

MÍRIAM TIMIRAOS DÍAZ[§], *Department of Industrial Engineering, University of A Coruña, CTC, CITIC, 15403 Ferrol, A Coruña, Spain.*

FRANCISCO ZAYAS-GATO[§§], *Department of Industrial Engineering, University of A Coruña, CTC, CITIC, 15403 Ferrol, A Coruña, Spain.*

ESTEBAN JOVE[¶], *Department of Industrial Engineering, University of A Coruña, CTC, CITIC, 15403 Ferrol, A Coruña, Spain.*

JOSÉ-LUIS CASTELEIRO-ROCA[¶¶], *Department of Industrial Engineering, University of A Coruña, CTC, CITIC, 15403 Ferrol, A Coruña, Spain.*

HÉCTOR QUINTIÁN[‡], *Department of Industrial Engineering, University of A Coruña, CTC, CITIC, 15403 Ferrol, A Coruña, Spain.*

HÉCTOR ALAIZ-MORETÓN[‡‡], *Department of Electrical and Systems Engineering, University of León, 24007 León, Spain.*

JOSÉ LUIS CALVO-ROLLE[#], *Department of Industrial Engineering, University of A Coruña, CTC, CITIC, 15403 Ferrol, A Coruña, Spain.*

[*]E-mail: alvaro.michelena@udc.es
[**]E-mail: mgaro@unileon.es
[†]E-mail: jose.aveleira@unileon.es
[††]E-mail: david.yeregui.marcos@gmail.com
[§]E-mail: miriam.timiraos.diaz@udc.es
[§§]E-mail: f.zayas.gato@udc.es
[¶]E-mail: esteban.jove@udc.es
[¶¶]E-mail: jose.luis.casteleiro@udc.es
[‡]E-mail: hector.quintian@udc.es
[‡‡]E-mail: hector.moreton@unileon.es
[#]E-mail: jlcalvo@udc.es

Abstract

This paper aims to enhance security in IoT device networks through a visual tool that utilizes three projection techniques, including Beta Hebbian Learning (BHL), t-distributed Stochastic Neighbor Embedding (t-SNE) and ISOMAP, in order to facilitate the identification of network attacks by human experts. This work research begins with the creation of a testing environment with IoT devices and web clients, simulating attacks over Message Queuing Telemetry Transport (MQTT) for recording all relevant traffic information. The unsupervised algorithms chosen provide a set of projections that enable human experts to visually identify most attacks in real-time, making it a powerful tool that can be implemented in IoT environments easily.

Keywords: Beta Hebbian Learning, t-SNE, ISOMAP, IoT, MQTT, cyberattack.

1 Introduction

The concept of the Internet of Things (IoT) alludes to the implementation of internet connectivity to everyday objects. This is achieved through sensors (to capture information) and actuators (to perform certain actions) to interact with the environment.

In the household, IoT is commonly applied to lighting, home appliances and thermostats to increase automation in routine tasks and improve overall well-being. The favorable reception to IoT has prompted Tech giants like Google, Phillips or Amazon [35] to make important investments to develop new connected devices.

When it comes to the industrial sector, IoT is usually included in the broader term 'Industry 4.0' [18]. Specifically, it refers to real-time management and operation of automation-controlled systems through remote monitoring from a connected client device (PC, smartphone or tablet with internet access).

According to the yearly Internet Report from Cisco [9], there is an increase in the average number of connected devices and overall connections per household due to the proliferation of devices implementing IoT capabilities such as intelligent thermostats, surveillance systems, healthcare telemetry, transportation or e-commerce tracking tools. By 2023, IoT device connections are set to account for over 50% of the total new accesses to the internet.

IoT devices, regardless of their field of application, share certain features: they implement small-sized electronics in order to be energetically efficient and be able to be easily integrated. Consequently, their computing capabilities are rather limited. Additionally, the protocols also differ from the ones implemented in traditional networks. The primary objectives are efficiency, scalability, and the ability to communicate in real time. This uniqueness in IoT devices introduces new cybersecurity challenges in the form of novel attack vectors specific to them.

An attack vector or threat in the cybersecurity field alludes to any activity exploiting security breaches in a system in order to cause a negative effect. The two main cybersecurity risk sources are humans and the environment, according to [26].

One of the most extended methods of exploiting attack vectors is the generation of botnets (in this case, IoT devices infected to be used as part of an attack). A good recent example is dark_nexus, which, according to the analysis by [5], is specific to IoT and has compromised over 1370 devices so far.

Apart from code-injections to create botnets in order to attack a certain system as stated above, the existing vulnerabilities are exploited to compromise the IoT environments with the purpose of

Vol. 32, No. 2, © The Author(s) 2024. Published by Oxford University Press.

Advance Access published 20 March 2024 https://doi.org/10.1093/jigpal/jzae013

gaining control of it, obtaining information or simply shutting it down. The following classification introduces the main types of IoT attacks not related to a specific software or manufacturer, according to [11, 20]. They can be implemented by taking advantage of the IoT related protocols, the IoT system itself or the associated network [30]:

- Sniffing: Revolving around capturing the information traffic in a network to unveil the protocols, ports and devices. This is usually the first step in a more sophisticated attack because it brings important information regarding potential vulnerabilities in the different components of the system.
- Denial of Service (DoS): The underlying concept is the presence of an attacker sending large amounts of traffic to the system until overflowing it, causing a halt in the services. In its distributed variant, the traffic in generated simultaneously from several nodes in the network as in the botnet DDoS. DoS attacks are especially relevant in IoT systems due to their limited computation capacity and real-time nature, making the especially prone to overflowing.
- MitM (Man in the Middle): the attacker analyses the traffic between two parties, e.g. between a sensor and the server. Subsequently, it positions itself between the two parties maintaining the existing connection unaltered. By doing so, the attacker gains access to all the information exchanged between the sensor and the server and can even alter it with false information [19].
- Sybil Attack: the attacker generates a fictitious device belonging to the IoT system, which can interact with it and gain access to information. This kind of attack is possible because the computation capacity of IoT systems usually does not allow the implementation of Public Key-based cryptographic primitives to identify unauthorized devices [2].

One possible solution to increase security in IoT systems without changing the pre-existing values is to integrate an Intrusion Detection System (IDS) because its mechanism is based on network traffic analysis without altering any component. Just by analysing the information flow, an IDS system can detect occurring attacks in real time [34]. The above-mentioned characteristic constitutes a very relevant advantage for the introduction of IDS-based anomaly detection in IoT systems [3]. Usually, IDS utilize classification models created with AI. The training is performed with big sets of structured and organized data.

Previous works of the authors addressed the development of an Intrusion Detection System where intelligent models were deployed. These models were based on the implementations of several machine learning and deep learning algorithms with satisfactory results [1, 6, 7, 12, 15]. Being the experiments, for modeling purposes, made over the same dataset, which will be presented in the 2 section.

Building on this idea, the current article introduces a proposal for visualization of the IoT network behavior when the MQTT IoT protocol is threatened. With the aim to get this purpose, three different methods are implemented, Beta Hebbian Learning (BHL), t-distributed Stochastic Neighbor Embedding (t-SNE) and ISOMAP algorithm. The output of these methods will make it possible that an expert in network communications visualizes how the network is working, being this, a complement for automatic IDS operation.

The rest of this paper is organized as follows: section 2 presents the case of the study and the testing network environment, following section 3 reviews and presents the methods and algorithms applied in this research, to present in section 4 the experiments conducted and the results obtained. Finally, the conclusions of this study and the proposals for future work are stated in section 5.

2 Case study

The MQTT protocol belongs to the publish-subscribe category and it is very suitable for M2M (Machine to Machine) communication where interaction between machines or devices is without the need for human intervention. MQTT is typically utilised to connect small devices with limited broadband capacity in IoT [29] and industrial [24] environments. The protocol uses a star-shaped architecture with a central node, called a 'broker', which acts as a server and is responsible for network management and real-time message transmission. MQTT communication is based on topics that can be created and published by any client. Clients can subscribe to topics to receive all messages related to that topic, and it is possible to subscribe to multiple topics simultaneously. Communication can be one-to-one or one-to-many, and topics have a hierarchical structure separated by the symbol '/'. The complexity of the system is handled by the broker. MQTT uses TCP (Transmission Control Protocol) as a transport protocol and TLS/SSL (Transport Layer Security/Secure Sockets Layer) for security, but does not specify any particular network or routing technique. Typically, messages up to 256 MB have a 2-byte header.

As previously mentioned, datasets are crucial for training machine-learning models used in IDS systems. For effective training, the datasets must include both normal and attacked traffic. In the case of IoT environments and the MQTT protocol, there are several fitting and up-to-date datasets available, including:

- MQTT-IoT-ids2020 [13], which is generated through a simulated MQTT architecture comprised of 12 sensors emitting random messages. The dataset includes network scans and brute force attacks performed by the aggressor to decipher access credentials.
- Bot-IoT [17], developed as a benchmark consisting of several virtual machines running on different operating systems, firewalls and the Node-RED visual programming tool. The dataset includes DoS and DDoS attacks, with MQTT traffic generated by the Node-RED tool simulating a weather station that was subject to various attacks [21].
- MQTT-set, a dataset specific to the MQTT protocol, was developed by the University of Genoa, Italy in 2020 [31] using the IoT-Flock tool to generate traffic that can mimic networks and devices of the MQTT and CoAP protocols. The dataset includes a malicious element that launches DoS attacks and generates malformed traffic, creating large amounts of exceptions.

In machine learning training models for IDS systems, datasets play a crucial role, and it is mandatory for these datasets to include two types of marked traffic: normal and under attack. However, the previously introduced datasets have two main limitations: they are centered around common attacks that can occur on any type of network, such as botnets or denial of service, and they do not generate real traffic, but rather, simulated.

To overcome these limitations, the authors of this article propose a specific attack that targets the particular vulnerabilities of the MQTT protocol through a Sybil attack tailored to the present case. The attack vector exists when there is no authentication to access the broker due to the limited computing capacity of certain devices. By scanning port 1833, it is possible to determine which servers are using the MQTT protocol and which are available. The Shodan Scanner, for instance, can be used to discover many unprotected brokers. Once the server is identified, it is possible to check what topics are managed by the broker using the special character '#' [4, 16]. This vulnerability could be exploited by an external attacker to know the active topics, subscribe and capture important information, or even publish false instances.

TABLE 1. Variables used as inputs

frame.time delta	frame.time delta displayed	qtt.willtopic len
frame.time epoch	frame.time invalid	frame.time relative
ip.src	ip.dst	tcp.srcport
tcp.dstport	eth.src	eth.dst
frame.cap len	frame.coloring rule.name	frame.coloring rule.string
frame.comment	frame.comment.expert	frame.encap type
frame.file off	frame.ignored	frame.incomplete
frame.interface id	frame.interface name	frame.len
frame.link nr	frame.marked	frame.md5 hash
frame.number	frame.offset shift	mqtt.clientid
mqtt.clientid len	mqtt.conack.flags	mqtt.conack.flags.reserved
mqtt.conack.flags.sp	mqtt.conack.val	mqtt.conflag.cleansess
mqtt.conflag.passwd	mqtt.conflag.qos	mqtt.conflag.reserved
mqtt.conflag.retain	mqtt.conflag.uname	mqtt.conflag.willflag
mqtt.conflags	mqtt.dupflag	mqtt.hdrflags
mqtt.kalive	mqtt.len	mqtt.msg
mqtt.msgid	mqtt.msgtype	mqtt.passwd
mqtt.passwd len	mqtt.proto len	mqtt.protoname
mqtt.qos	mqtt.retain	mqtt.sub.qos
mqtt.suback.qos	mqtt.topic	mqtt.topic len
mqtt.username	mqtt.username len	mqtt.ver
mqtt.willmsg	mqtt.willmsg len	mqtt.willtopic

To generate a legitimate dataset to be used in the IoT system implementing the MQTT protocol and collect the attack, a benchmark over a WLAN (Wireless Local Area Network) is developed to obtain real traffic. The broker is programmed in node.js using the 'Aedes' library, and two 'ESP8266' chipsets are placed and connected to sensors and actuators via their GPIO (General Purpose Input/Output) pins. Finally, both smartphones and computers are used as clients to browse on the internet and interact with the IoT system. All the traffic generated in this environment is captured by a router with the 'OpenWRT' Operating System.

For obtaining a dataset including both normal and under-attack traffic, all the information generated by the router is collected in Pcap format and subsequently dissected, taking those 38 fields belonging to the MQTT protocol from the 'Wireshark Display Filter Reference'. 28 fields common to all the frames of the gathered traffic are selected and offer relevant information in all cases, among these fields are the system times and the relative time of collating Table 1. Finally, each frame is marked as 'normal' or 'under attack', considering the timing of each attack.

The Sybil attack is performed from an MQTT Mosquito client subscribed to the topic '#' so that all the information generated by the other clients and the topics to be used is obtained. The attacker can subsequently connect with the same Mosquito client to publish false information regarding both the relay and the sensor. The generated dataset in CSV (Coma Separated Values) format is comprised of 78995 normal instances and 80893 under-attack samples for a total of 65 variables representing 1898 attacks. It is available at https://joseaveleira.es/dataset

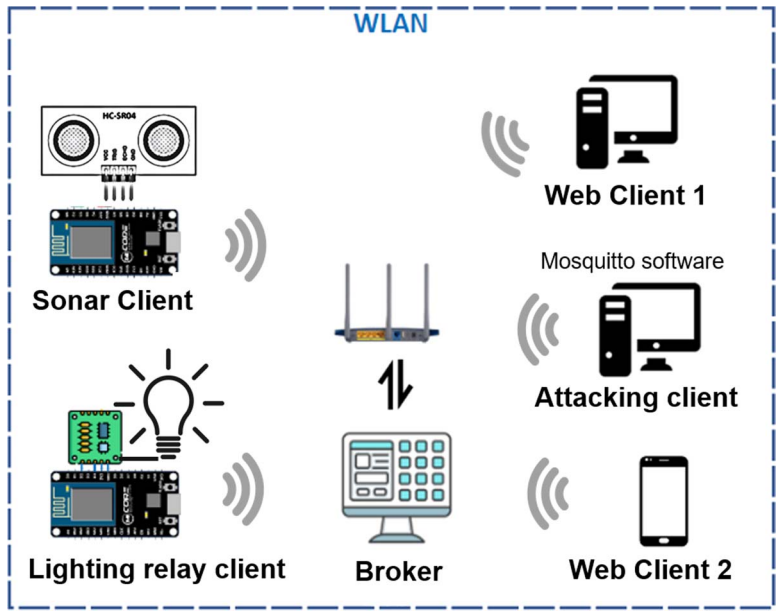

FIGURE 1. MQTT network environment.

After an initial analysis of the original dataset, a total of 21 variables with constant values were identified and therefore removed from the original dataset, finally obtaining a dataset with 41 variables. In all cases, categorical variables have been coded as numeric ones.

3 Methods as projection model for cyberattacks visualization

In order to obtain a visualisation of the behaviour of MQTT attack, a set of techniques are introduced, being some of them applied to dataset explained in the section 2.

Unsupervised artificial neural networks have numerous applications, with data projection or visualization being particularly helpful for human experts analysing a dataset's internal structure. This is achieved by projecting the data onto informative axes or generating maps depicting the dataset's inner structure. Exploratory Projection Pursuit (EPP) is a technique commonly used for this type of data visualization [8, 10, 22], allowing experts to inspect structures visually by projecting data onto a low-dimensional subspace. Projectionist techniques have been successfully employed for intrusion detection networks in prior research [23, 27, 28, 33], serving as a useful tool for detecting anomalous situations and understanding attacks. Such techniques provide a clear visualization of a network's internal data structure that can be interpreted by human experts. The following section details the methods utilized to generate the projection model for visualizing cyberattacks.

3.1 Beta Hebbian Learning algorithm

While other EPP algorithms have previously yielded satisfactory results, a novel technique called Beta Hebbian Learning (BHL) [22] has recently been found to significantly outperform commonly

used methods like PCA, MLHL, and CMLHL. BHL is an unsupervised EPP Artificial Neural Network that employs Beta distribution as part of its weight update process to project high-dimensional datasets onto low-dimensional (usually two-dimensional) subspaces for information extraction. Compared to other exploratory methods, BHL offers a clearer representation of the data's internal structure. The learning rule of BHL involves using Beta distribution to update the weights by aligning the Probability Density Function (PDF) of the residual (e) with the distribution of the dataset. The residual refers to the difference between the input and output feedback through the weights (4). The optimal cost function can be determined by knowing the residuals' PDF.

Therefore, the residual (e) can be expressed by 5 in terms of Beta distribution parameters ($B(\alpha \; and \; \beta)$):

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

where α and β control the PDF shape of the Beta distribution, e is the residual, x are the 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 (Eq. 2,):

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

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

$$Feed-forward : y_i = \sum_{j=1}^{N} W_{ij}x_j, \forall i \tag{3}$$

$$Feedback : e_j = x_j - \sum_{i=1}^{M} W_{ij}y_i \tag{4}$$

$$Weightupdate : \Delta W_{ij} = \eta(e_j^{\alpha-2}(1-e_j)^{\beta-2}(1-\alpha+e_j(\alpha+\beta-2)))y_i \tag{5}$$

where η is the learning rate

3.2 t-distributed Stochastic Neighbor Embedding (t-SNE)

T-distributed Stochastic Neighbor Embedding or t-SNE is a statistical, non-linear method for high-dimensional data analysis and visualization into lower-dimensional space[32].

t-SNE approach is to transform high-dimensional data points into joint probabilities, which in turn are utilized to minimize the Kullback–Leibler divergence (KLd) [14] and ultimately obtain low-dimensional embeddings. Initially, the algorithm obtains the joint probabilities for the data points (similarity) and assigns the similarities between the data points. Finally, t-SNE generates the representation of the data points on lower dimensions based on the probability distribution and iterates until the lowest KLd is reached. t-SNE presents a non-convex cost function, so it is possible to obtain diverse outcomes for each iteration. On the other hand, one big advantage is the fact that, contrary to other methods like PCA, t-SNE presents a much better performance retaining low pairwise separation, and therefore better visualization is obtained. This is due to its non-linear behaviour approach to data.

3.3 ISOMAP

The Euclidean separation for nonlinear manifolds holds when the neighborhood structure can be considered linear; otherwise, it might not be trustable. It is possible to accurately estimate the separation between two given points by following the manifold to calculate the distance.

Considering a 2-D example, the system will reduce the data to 1-D by means of Euclidean separations and approximate geodesic distances. In such cases, the items with a high degree of separation are mapped imperfectly, while those points suitable to be considered to repose in a linear manifold yield a correct value. With Isomap, the distance between two points is defined by the graph separating them. Therefore, Euclidean separation is not a suitable option for separation estimation in non-linear manifolds, and geodesic distance constitutes a more accurate choice [36]. In other words, Isomap uses local information to generate a resemblance matrix for eigenvalue decomposition. Subsequently, the euclidean metrics generate the neighborhood graph to finally calculate the geodesic separation for any two points through the shortest path using the graph separation. As a consequence, both the general and local structure of the dataset are estimated in the low-dimension embedding.

4 Experiments and results

Following, the experimental setup and results of the implementation of Beta Hebbian Learning, t-SNE, ISOMAP techniques are addressed.

For the three techniques implemented normalization procedure has been followed for each variable between -1 and 1 [22, 25]. After that, the best projections from a visual point of view were presented, using different colors for attack and standard traffic. Following the experimental setup for each technique and the results as a set of graphics protections are presented:

4.1 Beta Hebbian Learning algorithm

The figure depicts the first three components of the dataset projected onto a new subspace, with normal samples represented in green and attacks in red. To generate these projections, the BHL algorithm was used with a parameter setting of 5000 iterations, a learning rate of 0.001 and $\alpha = 4$, $\beta = 3$.

In the 3D projection, normal samples are distinctly separated from attacks, with a few isolated samples and a region in the lower right corner of the image making differentiation difficult. However, rotating the view and zooming in allows for a clearer separation of the two types, as demonstrated in Figure 3.

4.2 t-distributed Stochastic Neighbor Embedding (t-SNE)

The parameters for t-SNE have been ”perplexity”=30. This makes reference to the nearest neighbor number typical in the manifold family algorithms. ”Early exaggeration”=12.0. Thanks to this, it is possible to control the space between clusters and their density. Finally, the number of iterations has been 1000, and the learning is equal to 200. Figure 4 shows the a 3D data protections. In this case, it is impossible to differentiate how the attacks and normal traffic are situated, even changing visualization angles. Finally, it is not possible to assess the IoT network behavior.

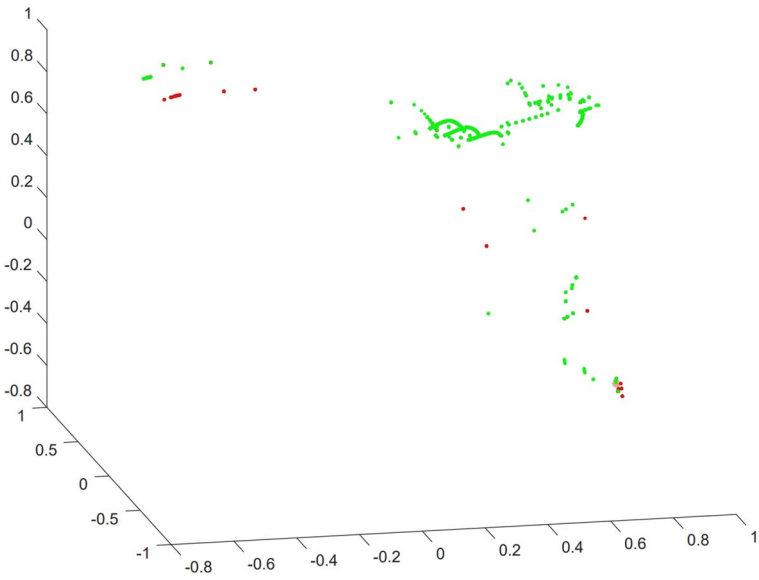

FIGURE 2. 3D BHL 3 first components projection.

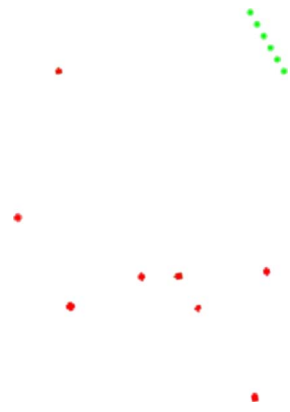

FIGURE 3. 2D BHL 3 first components projection, rotated and zoom.

4.3 ISOMAP

The last technique implemented has been ISOMAP, which is a technique from the manifold algorithms family. This is the most complex algorithm applied in this work, both from a computational and experimental setup perspective, due to the high number of parameters involved. Therefore, the set of parameters is displayed as a set of items:

- n_neighbours=5: number of neighbors for each point considered.
- eigen_solver=”auto”, to get the optimal solver.
- path_method=”auto”, to get automatically the best algorithm.

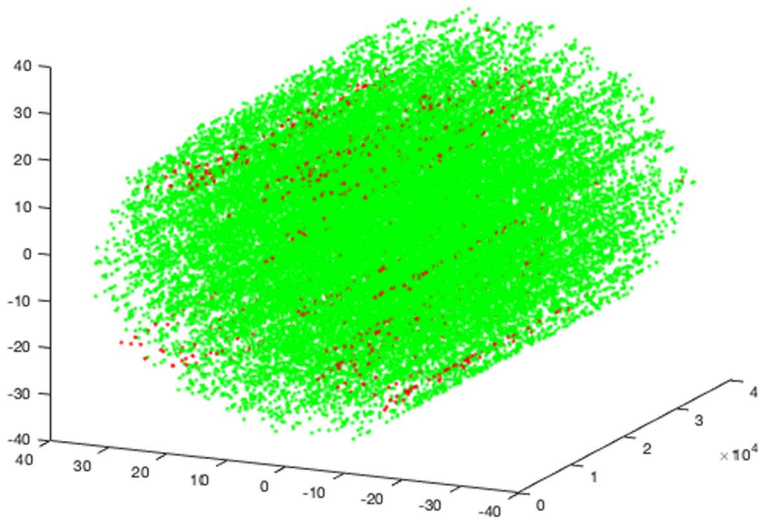

FIGURE 4. 3D t-SNE 3 first components projection example.

- path_method=”auto”, with the aim of getting automatically the best algorithm between ”Floyd-Warshall” and ”Dijkstra’s”.
- metric=”minkowski”, for calculating the distance between instances in a feature array.
- p=2, for using euclidean distance in Minkowski metric.

Graphical representation of ISOMAP, figure 5, shows that in this case, there is the possibility of knowing how the attacks and normal traffic are situated in different places. So, with three components, ISOMAP is able to capture the network behavior.

5 Conclusions and future works

The importance of security issues in IoT devices is continuously increasing, making it crucial to have a visual tool that can present the internal structure and behavior of networks. This tool offers distinct advantages, especially in detecting and classifying new attack modalities. Industrial processes and communication protocols, such as the MQTT protocol, are particularly vulnerable to attacks due to their features. Therefore, early detection of attacks is essential to ensure the resilience of these processes.

This study explores the application of three unsupervised machine learning techniques—Beta Hebbian Learning (BHL), t-distributed Stochastic Neighbour Embedding (t-SNE) and ISOMAP—on a dataset based on several attacks in an IoT environment utilizing the MQTT communication protocol. The results show intriguing projections of the dataset’s internal structure, making it possible to distinguish between normal network behavior and attack instances, with the Beta Hebbian Learning approach being the best. The main advantage of the BHL approach is that it provides a simple tool for human experts to extract knowledge about network operations during an attack. As a result, new attack modalities can be identified and classified without relying on any prior information.

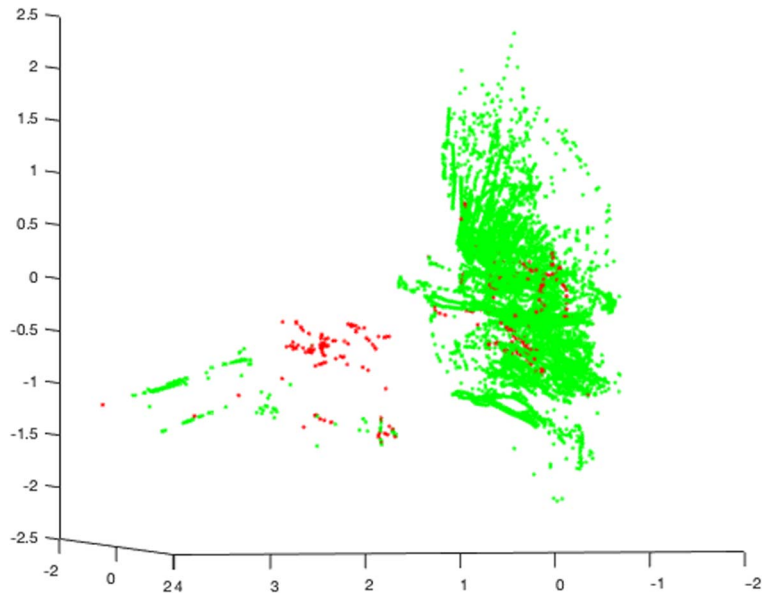

FIGURE 5. 3D ISOMAP 3 first components projection example.

Future research will focus on evaluating the BHL and ISOMAP algorithm's performance on a CoAP (Constrained Application Protocol) dataset based on the set of attacks developed by the authors in current works. Additionally, there are plans to integrate supervised learning algorithms with the BHL technique to create a more resilient system for preventing new threats.

Acknowledgements

CITIC, as a Research Center of the University System of Galicia, is funded by Consellería de Educación, Universidade e Formación Profesional of the Xunta de Galicia through the European Regional Development Fund (ERDF) and the Secretaría Xeral de Universidades (Ref. ED431G 2019/01).

Álvaro Michelena's research was supported by the Spanish Ministry of Universities (https://www.universidades.gob.es/), under the 'Formación de Profesorado Universitario' grant with reference FPU21/00932.

Spanish National Cybersecurity Institute (INCIBE) and developed the Research Institute of Applied Sciences in Cybersecurity (RIASC).

This initiative is carried out within the framework of the funds of the Recovery, Transformation and Resilience Plan, financed by the European Union (Next Generation), the project of the Government of Spain that outlines the roadmap for the modernization of the Spanish economy, the recovery of economic growth and job creation, for the solid, inclusive and resilient economic reconstruction after the COVID-19 crisis, and to respond to the challenges of the next decade.

Míriam Timiraos's research was supported by the Xunta de Galicia (Regional Government of Galicia) through grants to industrial Ph.D. (http://gain.xunta.gal), under the Doutoramento Industrial 2022 grant with reference: 04_IN606D_2022_2692965.

Funding for open access charge: Universidade da Coruña/CISUG

References

[1] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García and C. Benavides. Multiclass classification procedure for detecting attacks on mqtt-iot protocol. *Complexity*, **2019**, 1–11, 2019.

[2] S. A. Alamànd and D. De. Analysis of security threats in wireless sensor network. *International Journal of Wireless Mobile Networks (IJWMN)*, **6**, 35–46, 2014.

[3] M. A. Alsoufi, S. Razak, M. M. Siraj, I. Nafea, F. A. Ghaleb, F. Saeed and M. Nasser. Anomaly-based intrusion detection systems in iot using deep learning: a systematic literature review. *Applied Sciences*, **11**, 2021.

[4] S. Andy, B. Rahardjo and B. Hanindhito. Attack scenarios and security analysis of mqtt communication protocol in iot system. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 1–6, 2017.

[5] L. Arsene. *New Dark Nexus Iot Botnet Puts Others to Shame*. BitDefender, 2020.

[6] J. Aveleira-Mata, E. Jove, D. Y. M. del Blanco, M. T. G. Ordás, F. Zayas-Gato, Á. Michelena, J.-L. Casteleiro-Roca, H. Quintián, H. Alaiz-Moretón and J. L. Calvo-Rolle. Detection of denial of service attacks in an mqtt environment using a one-class approach. In *Computational Intelligence in Security for Information Systems Conference*, pp. 84–93. Springer, 2021.

[7] J. Aveleira-Mata, Á. L. Muñoz-Castañeda, C. Benavides-Cuellar, J. A. Benítez-Andrades, M. T. García-Ordás, C. Benavides-Cuellar and J. Alberto. Prototipo de IDS para detección de intrusiones con modelos de machine learning en sistemas IoT de la Industria 4.0 (IDS prototype for intrusion detection with machine learning models in IoT systems of the Industry 4.0). *Dyna*, **96**, 270–275, 2021.

[8] A. Berro, S. L. Marie-Sainte and A. Ruiz-Gazen. Genetic algorithms and particle swarm optimization for exploratory projection pursuit. *Annals of Mathematics and Artificial Intelligence*, **60**, 153–178, 10, 2010.

[9] Cisco. *Cisco Annual Internet Report—Cisco Annual Internet Report (2018–2023) White Paper*, 2020.

[10] E. Corchado and C. Fyfe. Connectionist techniques for the identification and suppression of interfering underlying factors. *IJPRAI*, **17**, 1447–1466, 2003.

[11] J. Deogirikar and A. Vidhate. Security attacks in iot: a survey. *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC*, **2017**, 32–37, 2017.

[12] M. T. García-Ordás, J. Aveleira-Mata, J.-L. Casteleiro-Roca, J. L. Calvo-Rolle, C. Benavides-Cuellar and H. Alaiz-Moretón. Autoencoder latent space influence on iot mqtt attack classification. In *Intelligent Data Engineering and Automated Learning—IDEAL 2020*, C. Analide, P. Novais, D. Camacho and H. Yin., eds, pp. 279–286. Springer International Publishing, Cham, 2020.

[13] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis and X. Bellekens. *Machine Learning Based Iot Intrusion Detection System: An Mqtt Case Study (Mqtt-Iot-ids2020 Dataset), 9*, 2021.

[14] S. Ji, Z. Zhang, S. Ying, L. Wang, X. Zhao and Y. Gao. Kullback–Leibler divergence metric learning. *IEEE Transactions on Cybernetics*, 2020.

[15] E. Jove, J. Aveleira-Mata, H. Alaiz-Moretón, J.-L. Casteleiro-Roca, D. Y. M. del Blanco, F. Zayas-Gato, H. Quintián and J. L. Calvo-Rolle. Intelligent one-class classifiers for the development of an intrusion detection system: the mqtt case study. *Electronics*, **11**, 422, 2022.

[16] D. Kant, A. Johannsen and R. Creutzburg. Analysis of IoT security risks based on the exposure of the MQTT protocol. *IEEE Access*, **33**, 96-1–8, 2021.

[17] N. Koroniotis, N. Moustafa, E. Sitnikova and B. Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, **100**, 779–796, 2019.

[18] J. Lee and A. Kao. Industry 4.0 factory in big data environment. tec. News. *HARTING's Technology Newsletter*, **26**, 8–9, 2014.

[19] A. Mallik. Man-in-the-middle-attack: understanding in simple words. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, **2**:109, 1 2019.

[20] M. Nawir, A. Amir, N. Yaakob and O. B. Lynn. Internet of things (iot): taxonomy of security attacks. In *2016 3rd International Conference on Electronic Design, ICED 2016*, pp. 321–326, 2017.

[21] J. M. Peterson, J. L. Leevy and T. M. Khoshgoftaar. A review and analysis of the bot-iot dataset. In *Proceedings—15th IEEE International Conference on Service-Oriented System Engineering, SOSE 2021*, pp. 20–27, 8, 2021.

[22] H. Quintián and E. Corchado. Beta hebbian learning as a new method for exploratory projection pursuit. *International Journal of Neural Systems*, **27**, 1750024–16, 2017.

[23] H. Quintián, E. Jove, J. L. Casteleiro-Roca, D. Urda, Á. Arroyo, J.,. L. Calvo-Rolle, Á. Herrero and E. Corchado. Beta-hebbian learning for visualizing intrusions in flows. In *13th International Conference on Computational Intelligence in Security for Information Systems, CISIS 2020, Burgos, Spain, September 2020*, Á. Herrero, C. Cambra, D. Urda, J. Sedano, H. Quintián and E. Corchado., eds. Advances in Intelligent Systems and Computing, vol. 1267, pp. 446–459. Springer, 2020.

[24] K. Ramamoorthy, S. Karthikeyan and T. Chelladurai. An investigation on industrial internet of things for mission critical things in industry 4. 0 2. Literature review. *Seybold Report*, **15**, 3294–3300, 2020.

[25] F. Ratle, A.-L. Terrettaz-Zufferey, M. Kanevski, P. Esseiva and O. Ribaux. Learning manifolds in forensic data. In *International Conference on Artificial Neural Networks*, pp. 894–903. Springer, 2006.

[26] J. J. Jaccard and S. Nepal. A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, **80**, 973–993, 2014.

[27] R. Sánchez, Á. Herrero and E. Corchado. Clustering extension of MOVICAB-IDS to distinguish intrusions in flow-based data. *Logic Journal of IGPL*, **25**, 83–102, 2017.

[28] 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 IGPL*, **25**, 54–66, 2017.

[29] P. Sethi and S. R. Sarangi. Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer*, **2017**, 2017.

[30] J. Tournier, F. Lesueur, F. Le Mouël, L. Guyon and H. Ben-Hassine. A survey of iot protocols and their security issues through the lens of a generic iot stack. *Internet of Things*, **16**, 100264, 2021.

[31] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli and E. Cambiaso. Mqttset, a new dataset for machine learning techniques on mqtt. *Sensors 2020*, vol. 20, p. 6578, 2020.

[32] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, **9**, 2008.

[33] R. V. 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 IGPL*, **27**, 160–176, 2019.

[34] A. Waleed, A. F. Jamali and A. Masood. Which open-source ids? Snort, suricata or zeek. *Computer Networks*, **213**, 109116, 2022.

[35] X. Kuai, F. Wang, S. Jimenez, A. Lamontagne, J. Cummings and M. Hoikka. Characterizing dns behaviors of internet of things in edge networks. *IEEE Internet of Things Journal*, **7**, 7991–7998, 2020.

[36] H. Zha and Z. Zhang. Continuum isomap for manifold learnings. *Computational Statistics & Data Analysis*, **52**, 184–200, 2007.

Received 20 May 2022